

特開平9-320766

(43) 公開日 平成9年(1997)12月12日

(51) Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
H 0 5 B 37/02			H 0 5 B 37/02	G
F 2 1 V 33/00			F 2 1 V 33/00	Z
G 0 6 F 9/06	5 3 0		G 0 6 F 9/06	5 3 0 M

審査請求 未請求 請求項の数 6 O L 外国語出願 (全 281 頁)

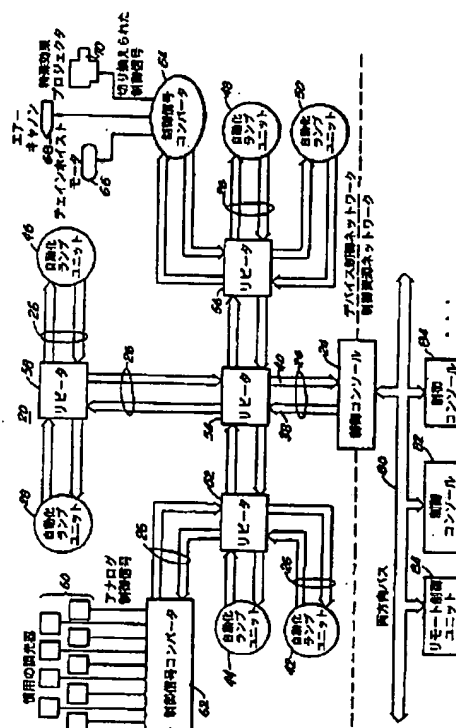
(21) 出願番号	特願平8-182598	(71) 出願人	591279412 ヴァリーライト インコーポレイテッド VARI-LITE INCORPORATED アメリカ合衆国 75247 テキサス州 ダラス リーガル ロウ 201
(22) 出願日	平成8年(1996)6月7日	(72) 発明者	ブルックス ダブリュ. テイラー アメリカ合衆国 75208 テキサス州 ダラス ローラエット ドライブ 2017
(31) 優先権主張番号	473, 150	(72) 発明者	トーマス イー. ウォルシュ アメリカ合衆国 75234 テキサス州 ダラス ヴェロニカ ロード 13206
(32) 優先日	1995年6月7日	(74) 代理人	弁理士 谷 義一 (外1名)
(33) 優先権主張国	米国 (US)		最終頁に続く

(54) 【発明の名称】 ライティングシステム

(57) 【要約】 (修正有)

【課題】 舞台装置の照明は数百の可動ランプユニット群や固定の数多のフロアランプ等を可動式はパン、チルト、色、光度、ビームサイズを個別に制御し、固定式は色切換えや光度を制御し、各ランプが制御に対する応答が完全にできるようにする。

【解決手段】 制御コンソール24はデータリンク26を介してオペレートするよう接続し、データリンク26は両方向バス38、40を含む。バス38は制御コンソール24と、各ランプユニット28及び他の装置間にデータ通信を行い、バス40は照明システム20の各ランプユニットからコンソール24へのデータ通信を提供する。リンク26はリンク26を経て伝送されるコマンドの完全性保持のためリピータ52、54、56、58群を設ける。コンソール24はランプユニット制御だけでなくセット60のような調光器を制御し、制御信号コンバータ62はリンク26を介し調光器60に伝える。



【特許請求の範囲】

【請求項1】 照明システムのための分散制御システムであって、

A. 前記照明システムのオペレーションをディレクトするパラメタ制御入力を、指定されたフォーマットに応じてエンタする1つ以上の制御デバイスであって、前記パラメタ制御入力に結合されたデータ・プロセッサと、該データ・プロセッサに結合されたメモリとを含む制御デバイスと、

B. 前記パラメタ制御入力に関連するデータをストアしエディットしディスプレイする1つ以上のコンピューティング・デバイスであって、少なくとも、データプロセッサと、該データ・プロセッサに結合したメモリと、前記データ・プロセッサに結合したデータ・ディスプレイ・デバイスとを含むコンピューティング・デバイスと、

C. 個別のインタフェース・モジュールを制御する1つのデータ・プロセッサをそれぞれ含む1つ以上の負荷インタフェース・モジュールであって、データリンク信号をモニタし、少なくとも1つのデバイス制御データリンクをサポートする負荷インタフェース・モジュールと、

D. 前記制御デバイスと、前記コンピューティング・デバイスと、前記負荷インタフェース・モジュールとに接続されている制御資源データリンク・ネットワークと、

E. ビーム特性に係る複数の調整可能なパラメタを有する光ビームを生成する手段を有するとともに、前記パラメタ制御入力にตอบสนองして複数の前記パラメタを制御するドライブ手段を有するマルチプル・パラメタ・ランプユニットに、前記負荷インタフェース・モジュールを接続するための共通バスを有する少なくとも1つの前記デバイス制御データリンク・ネットワークとを備えたことを特徴とする分散制御システム。

【請求項2】 インテリジェント照明アシスタントを備えた分散制御照明システムであって、入力デバイスと、推論エンジンと、データ・レポジトリと、負荷インタフェース・モジュールと、マルチプル・パラメタ・ランプユニットと、ネットワークとを備え付けた分散制御照明システムをオペレートする方法であって、

オペレータ・パラメタ・コマンドを推論エンジンに入力デバイスを介して注入するステップと、前記オペレータ・パラメタ・コマンドを前記推論エンジンで評価するステップと、

前記データ・レポジトリに駐在する3次元モデルであって、分散制御照明システムの表現と、システム・オペレーティング環境を備えた3次元モデルと、前記オペレータ・パラメタ・コマンドを比較するステップと、

前記オペレータ・パラメタ・コマンドと前記3次元モデルに基づき、システム・パラメタ調整値を計算するステップと、

ポーズするステップと、

前記システム・コマンドを前記マルチプル・パラメタ・ランプに前記負荷インタフェース・モジュールと前記ネットワークを介して伝送するステップとを備えたことを特徴とする方法。

【請求項3】 請求項2において、前記3次元モデルの表現を、前記システム・パラメタ調整値により変更されたように、グラフィック・ディスプレイ・デバイスにディスプレイするステップをさらに備えたことを特徴とする方法。

【請求項4】 請求項2において、追加のオペレータ・コマンドを受信すると直ちに前記システム・コマンドをインプリメントするステップをさらに備えたことを特徴とする方法。

【請求項5】 オペレータ・コマンド入力デバイスと、推論エンジンと、データ・レポジトリと、グラフィック・ディスプレイ・デバイスとを備え付けたインテリジェント照明アシスタントを備えた分散制御照明システムのオペレーションをプログラミングする方法であって、オペレータ・コマンドを前記推論エンジンに入力デバイスを介して注入するステップと、前記オペレータ・コマンドを前記推論エンジンで評価するステップと、

前記データ・レポジトリに駐在する3次元モデルであって、分散制御システムの表現とシステム・オペレーティング環境とを備えている3次元モデルと、前記オペレータ・コマンドとを比較するステップと、システム・パラメタ調整値を前記オペレータ・コマンドと前記3次元モデルとに基づいて計算するステップと、前記オペレータ・コマンドによりオーダされたパラメタ調整値を得るためシステム・コマンドをコンポーズするステップと、前記3次元モデルの表現を、前記システム・パラメタ調整値により変更されたように、前記グラフィック・ディスプレイにディスプレイするステップとを備えたことを特徴とする方法。

【請求項6】 ネットワークにより接続された分散制御照明システムであって、オペレータ・コマンド入力デバイスと、システムデータ・レポジトリを有する汎用コンピュータと、負荷インタフェース・モジュールと、複数パラメタ・ランプユニットを備えた分散制御照明システムにおいて、インテリジェント照明アシスタントは、前記オペレータ入力デバイスと前記システムデータ・レポジトリに対話的に接続した推論エンジンと、システム・パラメタ調整値を計算するとともに、前記オペレータ・コマンドにตอบสนองしてシステム・コマンドを計算するように構成された推論エンジンであり、前記システム・コマンドを前記ネットワークを介して前記マルチプル・パラメタ・ランプユニットに伝送するためのインタフェース・モジュールに接続した推論エンジンと、前記データ・レポジトリは分散制御システムの表現とシ

ステム・オペレーティング環境とを備えた3次元モデルを保持し、

システム・パラメタ調整値の計算をインプリメントする結果と、前記3次元モデルへの前記オペレータ・コマンドに応答するシステム・コマンドとを、前記推論エンジンによりディスプレイするグラフィック・ディスプレイ・デバイスとを備えたことを特徴とする分散制御照明システム。

【発明の詳細な説明】

【0001】本出願は、1994年7月11日に付与された特許出願第08/273,262号の一部継続出願である。本出願は、1992年6月9日出願された特許出願第07/898,385号の一部継続出願である。特許出願第07/898,385号は1991年9月26日出願された特許出願第07/766,029号の一部継続出願である。

【0002】特許出願第07/766,029号は1990年7月19日出願された特許出願第07/555,946号の一部継続出願である。特許出願第07/555,946号は1988年9月22日出願された特許出願第07/249,225号（米国特許第4,980,806号）の一部継続出願である。特許出願第07/249,225号（米国特許第4,980,806号）は1987年11月12日出願された特許出願第07/120,743号の一部継続出願である。特許出願第07/120,743号は1986年7月17日出願された特許出願第06/887,178号（現在は放棄されている）の継続出願である。

【0003】

【発明の属する技術分野】本発明は、インテリジェント・リモート固定照明装置とインテリジェント・データ分配ネットワークを有するステージ照明システムに関し、特に、このようリモート固定照明装置とデータ分配ネットワークのモジュラー制御に関する。

【0004】

【従来の技術】高性能コンピュータ制御照明システム、例えば、米国特許第4,980,806号（発明者：Taylor他）で開示されているようなシステムは、数百というランプユニットでパラメタを同時に実行するため、マルチプル・パラメタ照明装置のリモート制御に関係付けをしたタスクを、大量のデータの通信を含めて、容易にハンドリングすることができる。しかし、システム全体のオペレーションに影響を与えることなく、個々のシステム・コンポーネントをアップグレードするか、あるいは、効果的な方法で置換することができるように、このようなシステムを益々柔軟にしようとする要求がある。さらに、異なる演技の種々の要件に適合させるため、必要に応じて、照明システムの種々の態様を再構成することができるように、ますます柔軟にすることはこのようなシステムで望まれている。

【0005】現代の照明システムの制御装置は、異なるデータ要件を有する別種のランプユニットを同時にサポートすることができなければならないことがしばしばあ

る。所望の照明効果を得るため、種々の制御デバイスとランプユニットを照明システムで利用することが、異なる製造業者により製造された照明装置であって、それぞれ、一意の通信プロトコルとデータ・フォーマットを有する照明装置を含めて、照明デザイナーに求められている。

【0006】

【発明が解決しようとする課題】しかし、不幸にも、空間が限られ、マンパワーが限られているので、単一の演技で用いられる異なるタイプの照明装置を制御するため、個別の照明制御装置を利用するのは実際的ではないことがしばしばある。その上、照明コンソールの多数のオペレータは、単一の演技で用いられる異なるタイプの照明装置を同様の方法で制御するため、標準のインタフェースを要求している。不幸にも、異なる製造業者により提供される種々の照明装置を制御することができる標準照明制御装置の開発は、別種の通信プロトコルと、照明装置の種々の製造業者により提供されるランプユニットと関係付けられたデータ・パラメタにより妨げられてきた。

【0007】従って、本発明の第1の目的は、独立であり、置換可能であり、再構成可能なモジュールにより構築される照明制御装置を提供することにある。

【0008】本発明の第2の目的は、1つ以上の制御デバイスと、別種の通信プロトコルと機能とデータフォーマットを有する複数のランプユニットとの間で、通信をコーディネートするインタフェース・システムを提供することにある。

【0009】本発明の第3の目的は、多数の異なるタイプのランプユニットを制御するための改善された手段を提供することにある。

【0010】本発明の第4の目的は、別種のランプユニットに対してパラメタの指示を容易にする照明制御装置を提供することにある。

【0011】本発明の第5の目的は、ランプユニットの各タイプに対して、別種のランプユニットに同様の方法でアクセスする標準ユーザ・インタフェースを有する照明制御装置を提供することにある。

【0012】

【課題を解決するための手段】本発明の1つのインプリメンテーションでは、複数のマルチプル・パラメタ・ランプユニットを制御するようにした照明システムに、制御装置が提供される。

【0013】この制御装置は、（1）パラメタ制御入力を受信するソース・インタフェース・プロセッサと、パラメタ制御入力を、照明システムを制御するシステム制御コマンドにエンコードするソース・インタフェース・プロセッサと協働する中央処理システムとを有する1次制御システムと、（2）1次制御システムの1つ以上のソース・インタフェース・プロセッサに結合された複数

ている。照明システム20の操作は制御コンソール24により指定される。制御コンソール24は照明システム20の照明効果をマニュアルでセットするか、あるいはシステム20を自動的に指令し、ストアされた照明キュー(cue)により決定された所望の照明効果を奏するようにサーブする。制御コンソール24は、ランプユニット群内の各ランプユニットにデータリンク26を介して接続されている。1つのランプユニットをランプユニット28として示す。

【0020】ランプユニット28のようなランプユニットは、それぞれ、コンソール24および各ランプユニット28の間で個別に通信ができるように一意のアドレスを有する。データリンク26は、さらに、ランプ30のようなベDESTAL・ランプと、ランプ32のようなフロア・ランプとに接続してある。各ランプ30および32は固定してあるが、これらのランプの光度は、コンソール24により生成されるコマンドにより制御することができる。操作時には、照明システム20により、ランプ28'のような可動ランプは、パンと、チルトと、色と、光度と、ビームサイズが各別に調整され、一方、ベDESTAL・ランプ30およびフロアランプ32はその光度が調整される。色切り換え機構34を追加して、ベDESTAL・ランプ30の色も調整することができる。舞台22を照明する一連の「キュー」を供給するように、照明システム20が操作される。照明システム20内のランプユニットは、それぞれ、各キューに対して必要な応答を個々に行うことができる。演技を完全なものにするには、所望の照明効果を奏するのに、虞らく、数百のキューを設定する必要がある。

【0021】図1に示す照明システム20は、少数のランプユニット、例えば、ランプユニット28のようなランプユニットを示す。

【0022】しかし、実際の舞台演技には、このようなランプユニットが数百個必要である。実際、屋外のロックコンサートでは最高1000個のランプユニットを使用する必要がある。数千という多くのコマンドを発生させ、1つの演技での各キューに対して、各ランプの各パラメータをドライブしなければならないことは容易に分かる。単一の演技で、数万のコマンドを必要とする可能性は極めて高い。

【0023】照明システム20による照明効果は、舞台演技に適正に同期させ、プログラムされた動作を効果的にしなければならない。ランプの1つが不正確に応答するか、あるいは応答しそこなった場合、視覚上の効果が壊される。従って、コンソール24により開始されるキューに、各ランプが適正に応答することが極めて重要である。

【0024】以前の自動化照明システムでは、照明システムの各光源に対してパラメータをそれぞれ設定するのに必要な各コマンドを、制御プロセッサが発生する必要が

あった。上述したように、制御プロセッサが数万ものコマンドを発生し、しかも、これらのコマンドがデータリンクを介して各ランプに正確に搬送されることが必要である。データ伝送になんらかのエラーがあった場合、ランプは誤って応答し、視覚上の効果が損なわれることになる。ステージ領域での電氣的環境は、極めて限定された区域で大量の電力消費があるため、音響装置および照明装置の両装置に対して、数多くの種類の干渉がある。この電氣的な干渉により、コンソールから各ランプへのデータ伝送が干渉され、各ランプが不適正に応答する。本発明による照明システム20は、これらの問題の多くを克服するように設計されている。一方、ある演技に対して任意の時点に同時に利用できるランプの個数は、ほとんど無制限に増加させることができる。

【0025】舞台装置の複数のアイテムを制御するのに利用される照明システム20の機能ブロック図を図2に示す。制御コンソール24はデータリンク26を介してオペレートするように接続してあり、複数のアイテムの舞台装置を制御する。データリンク26は両方向バス38および40を含む。データリンク26のバス38は、制御コンソール24と、照明システム20の各ランプユニットおよびその他の装置との間に、データ通信を提供する。バス40は、照明システム20の各ランプユニットから制御コンソール24へのデータ通信を提供する。

【0026】ランプユニット28に加えて追加したランプユニット42~50を図2に示す。

【0027】データリンク26は舞台22の領域内のかんりの区域に行きわたるように延ばしてある。データリンク26を経て伝送される電氣的コマンドの完全性(integrity)を保持するため、リピータ52, 54, 56, および58群を設けてある。リピータ52~58はデータリンク26を介して伝送されるデータを増幅したり分離する。詳細は後程説明する。

【0028】制御コンソール24は、ランプユニット28のような自動ランプを制御するだけでなく、セット60のような複数の慣用の調光器を制御するようにサーブする。データリンク26は制御信号コンバータ62に接続してある。制御信号コンバータ62は、データリンク26を介して受信されたデジタル信号を、セット60内の調光器のオペレーションをディレクトするためのアナログ制御信号に変換する。

【0029】制御コンソール24は、複数の慣用の色切り換え機構34、例えば、慣用のベDESTAL・ランプ30に取り付けられたゲル・スクローラのような機構を制御するのにも用いることができる。データリンク26は制御信号コンバータ63に接続されている。制御信号コンバータ63は、制御信号コンバータ62のように、データリンク26を介して受信されたデジタル信号をアナログ信号に変換する。このアナログ信号により色切り換え機構34のオペレーションが指定される。しかし、制

御信号コンバータ63は各制御チャンネルに対して光度および色パラメータを記憶するようにプログラムされ、さらに、各論理制御チャンネルに対して少なくとも2つのアナログ制御電圧出力を生成するようにプログラムされている。このような出力は、一方が慣用の調光器60の1つに印加され、他方が色切り換え機構34のうちの対応する色切り換え機構に印加される。

【0030】このように構成することにより、照明システムのプログラミングが簡単になる。というのは、適正に取り付けたランプユニットの光度および色パラメータを、オペレータが単一の制御チャンネルを選択して指定することができるからである。また、制御信号コンバータ63の光度制御出力から色制御出力を論理的に分離することにより、慣用のランプユニットの光度をフェードアウトしている間に色切り換え機構のポジションを保持するようにプログラムすることができる。このようにすると、グランドマスタ・フェーダを用いて照明システムをフェードアウトしている間に色が変化してしまうという煩わしさを除くことができる。

【0031】リピータ52-58はデータリンク26へのコネクションを拡張するようにサージする。これは「ファンアウト(fan out)」といわれる。

【0032】他の舞台アクション効果をコンソール24により付加的に制御することができる。例えば、データリンク26をリピータ56から制御信号コンバータ64に接続することができる。制御信号コンバータ64は制御信号を生成し、その制御信号によりチェインホイスト・モータ66と、エアーキャノン68と、特殊効果プロジェクト70のオペレーションを指定することができる。

【0033】制御コンソール24は、制御される舞台デバイスのセットとのインタフェースとして供されている。これらの舞台デバイスと、関係付けをした制御とは、「デバイス制御ネットワーク」といわれる。制御機能はコンソール24を含む複数の装置により提供される。この制御装置群は「制御資源ネットワーク」といわれる。この制御資源ネットワークは両方向バス80を含む。両方向バス80は制御コンソール82および84の間のデータ通信だけでなく、制御コンソール24の間にデータ通信も提供する。照明システム20の指示は、リモート制御装置81によるオペレーションによりリモート・ロケーションで行なわせることができる。リモート制御装置81も両方向バス80に接続されている。

【0034】制御コンソール24のフロントパネル84を図3に示す。フロントパネル84はランプユニット28のような自動化ランプをそれぞれ直接制御するか、あるいはランプユニットを全て自動制御するようにサージする。フロントパネル84は特定の照明セットアップにキュー番号を直接割当てるキースイッチ群86を含む。回転調整つまみ88、90、92および94群は特定の

ランプユニットまたはランプユニット群に対し色選択を行う。各回転調整つまみ96、98、100および102は、各ランプに対し、パン(pan)と、チルト(tilt)と、光度(intensity)と、ズーム(zoom)をそれぞれ制御する。キースイッチ104群はプリセット色選択の機能を有する。特定の照明キュー(cue)がストア(store)スイッチ106の操作によりコンソール・メモリに入力される。

【0035】グランドマスタ(grand master)フェード調整つまみ112により、全ての照明システム20光源に対し、全体的なフェーディング効果が一度に与えられる。暗転(black-out)スイッチ114はランプを一度に全部消す。クロスフェーダ(cross fader)116および118は、キューからキューへ移行する間に、光度を相対的に制御する。このようなキューの移行に含まれるキューの番号は、それぞれ、表示器120および122により表示される。キュー番号はキーパッド124によりコンソール24からエンタされる。“S”キーはキューをストアするためのものであり、“E”キーは新たなキューをエンタするためのものである。キーパッドから入力された現在のキューは、表示器126により表示される。キースイッチ群128は第1のキューのキュー番号をエンタするためのものである。キースイッチ群130は第2のキューのキュー番号をエンタするためのものである。

【0036】制御パネル84は、ランプユニットを直接マニュアル制御することができ、同様に、照明システム20に対するキューをストアしリコール(recall)することができる場合には、多くの形式をとることができる。

【0037】制御コンソール24の電気ブロック図を図4に示す。制御コンソール24の全体的な制御はCPU (central processing unit) 140により行うことができる。CPU 140として使用する代表的なマイクロプロセッサとしては、Motorolaの6800型がある。CPU 140はデータバス142およびアドレスバス144に接続してある。制御コンソール24はRAM (random access memory) 146とEPROM (electronically programmable read only memory) 148とを有する。メモリ146および148は両方ともデータバス142およびアドレスバス144に接続してある。コンソール24の他の要素のみならずCPU 140は、メモリ146および148に対し、書き込み、読み出しを行うことができる。

【0038】ハードディスク・ドライブ150は制御コンソール24に設けてあり、プログラムおよびデータの大容量記憶を行う。さらに、慣用のフロッピー・ディスクの読み出し書き込みを行うため、フロッピーディスク・ドライブ152が設けてある。制御装置154はハードディスク・ドライブ150を動作させるように接続され、制御コンソール24の残りの回路にデータバス

142およびアドレスバス144を介して接続してある。同様に、フロッピーディスク・ドライブ制御装置156は、フロッピーディスク・ドライブ152をオペレートするように接続され、さらに、データバス142およびアドレスバス144に接続してある。

【0039】コンソール・パネル84、すなわち、コンソール・パネル84に設けたスイッチと、光源と、光学エンコーダと、ポテンシオメータと、英数字ディスプレイ装置は、コンソールパネル84と、データバス142およびアドレスバス144とに接続したコンソール・パネル・インタフェース回路158を介してアクセスされる。

【0040】自動化ランプユニットとの通信は、DMA (direct memory access) 回路164と、通信制御装置166と、Manchesterエンコーダ168を用いて行われる。データバス142およびアドレスバス144は、ともに、DMA回路164および通信制御装置166に接続してある。また、DMA回路164と通信制御装置166の間で通信が行われる。Manchesterエンコーダ164は通信制御装置166と両方向で通信を行い、かつ、データリンク26との間でデータを送受信する。

【0041】図5、図6、図7および図8は、本発明のランプユニットの電子装置を示すブロック図である。図5はランプ・プロセッサと、メモリと、関係付けをしたコンポーネントを具体的に示す。図6、図7および図8は、光源ユニットの光ビームの特定のパラメータを具体的に駆動する回路を示すブロック図である。

【0042】図5を説明する。図5はランプ・プロセッサ・システム178を示す。データリンク26は送受信ラインを有し、これら送受信ラインに各増幅器180、182が接続してある。データリンク26の送受信ラインは、スイッチ184に接続してある。スイッチ184はソレノイド186により作動され、ソレノイド186は増幅器188により駆動される。スイッチ184は「ループバック」能力を提供し、データリンク26内の送受信ライン間を直接接続し、ランプユニット・プロセッサはデータリンク26を用いなくて自己検査を行えるようになっている。データリンク26の送受信ラインは、エンコーダ/デコーダ190 (Harris Semiconductor Products Division Model HD-6409) に入力されている。エンコーダ/デコーダ190はランプユニット・アドレスバス192およびランプユニット・データバス194に接続してある。

【0043】ランプ・プロセッサ・システム178は、ランプユニット内の全機能をディレクトするマイクロプロセッサ200を含み、かつ、光源ユニットのパラメータを制御する機構をドライブするコマンドを具体的に生成する。マイクロプロセッサ200はMotorola Model 68000が好ましい。これらのパラメータはパンと、チルトと、光度と、色と、ビームサイズを含む。マイクロプロセッ

サ200はアドレスバス192およびデータバス194に接続してある。ランプ・プロセッサ・システム178は、さらに、RAMおよびEPROMメモリ202を含む。種々のパラメータを駆動して所望のステートにするプログラムと、これらのステートをどのようにさせるかを決定するキューとが、RAMおよびEPROMメモリ202にストアされる。マイクロプロセッサ200は、さらに、受信確認バス206および割り込みバス208を用いて、割り込みエンコーダ/受信確認回路204 (Motorola Model 68230) を介して割り込み信号を受信し、かつ、受信確認信号を送信するように接続してある。

【0044】ランプ・プロセッサ・システム178内の種々の回路素子のインタフェースおよびタイミングは、インタフェースおよびタイミング回路210 (Advanced Monolithics Model 9513) により提供されている。特定のランプユニットのIDは、ランプユニット一致回路212に含まれるつまみホイールの設定により判定される。このIDはインタフェースおよびタイマ回路210に入力される。電球電源214は、インタフェースおよびタイマ回路210にも伝送される種々の割り込みおよび受信確認状態を有する。

【0045】マイクロプロセッサ200はバス216を介してデコーダ218に伝送され、一連の制御信号を生成する。デコーダ218からの出力は制御信号群を備え、この制御信号群はデコーダ220に向けられ、さらに、ランプ・プロセッサ・システム178の多くの回路に制御コマンドとして分配される。制御信号群はデコーダ220により生成され、制御信号として、図6、図7および図8に示す具体的な制御回路に転送される。

【0046】データバス194を介して伝送されたデータは、バッファ228に供給される。バッファ228はデータを図6、図7および図8に示す種々のパラメータ制御回路に転送する。

【0047】バス206および208上の割り込みおよび受信確認信号は、ベクトル生成装置230に供給される。ベクトル生成装置230はバス232を介して伝送されるベクトル・ステートに応じて生成され、各データ線を介して、図6、図7および図8に示すパラメータ制御回路に伝送される。

【0048】割り込みバス208上の生成された割り込み信号は、さらに、割り込み信号として図6、図7および図8に示すパラメータ制御回路に供給される。同様に、図6、図7および図8のパラメータ制御回路により生成された受信確認信号は、バス206を介して割り込みエンコーダ/受信確認回路204に伝送される。

【0049】データバス194は、さらに、バッファ238に接続されており、バッファ238はDMA制御回路 (direct memory access controller; DMAC) 240 (Motorola Model 68440) とバッファ242の入力端子とにデータを伝送している。バッファ242の出力はア

ドレスバス192に供給される。ハンドシェイク制御信号はDMA制御回路240およびマルチプロトコル制御装置246の間で授受され、マイクロプロセッサ200との間で、データの高速通信を同期させる。

【0050】制御バス244はDMA制御回路240およびマルチプロトコル通信制御装置246の間の両方向コネクションとしてサブされている(Rockwell International Corp. Model 68561)。エンコーダ/デコーダ190は受信データおよび受信クロックをマルチプロトコル制御装置246に供給する。送信データおよび送信クロックはマルチプロトコル制御装置246からエンコーダ/デコーダ190に渡す。種々の制御信号はマルチプロトコル制御装置246およびエンコーダ/デコーダ190の間で交換される。

【0051】割り込み発生イベントが生じた場合、マルチプロトコル制御装置246はマイクロプロセッサ200に向けられた割り込み出力をアサートする。マイクロプロセッサ200からの割り込み受信確認に応答して、マルチプロトコル制御装置246は割り込みベクトルをデータバス194に出力する。慣用の方法では、マイクロプロセッサ200は一時的に処理に割り込み、割り込みをサービスする。

【0052】マルチプロトコル制御装置246はシリアルデータを伝送させ、パラレル・システム・データ入力の他に入力を受信する。識別されたタイプのマルチプロトコル制御装置246は、最高2 Mbit/sec のレートで、DMAデータ転送を行うことができる。このような性質を有する高速データ・ストリームにより、実質的な光源ユニット・キュー情報を極めて短い期間内でダウンロードすることができる。

【0053】エンコーダ/デコーダ190は、図4に示すように、通信制御装置166と協働してオペレートし、制御プロセッサ24によりシリアルに伝送されるデータのフォーマットまたはプロトコルを、ランプ・プロセッサ・システム178により受信可能なフォーマットに変換する。

【0054】ランプ・プロセッサ・システム178はクロックと、制御と、図示しない電源線(マイクロプロセッサ回路のオペレーションに通常必要とされている)のネットワークを含む。

【0055】ランプ・プロセッサ・システム178は、ランプユニット全体を初期化し、コンソールまたはストアされたキューからのマニュアル入力コマンドに回答して、パラメタ制御回路のオペレーションを指令し、ストアしたキューをメモリ202から制御コンソールに転送するようになっている。また、ランプ・プロセッサ・システム178はデータリンク26を介して受信したブロードキャスト・コマンドに回答し、ストア装置202からキューをリコールし、パラメタ制御回路のオペレーションを指令するようになっている。これらを図6、図7

および図8に示す。

【0056】図6を説明する。図6はパラメタ・ドライブ回路254を示す。パラメタ・ドライブ回路254はランプユニットで用いられるステッパモータをオペレートさせるようになっている。このようなステッパモータを用いて、例えば、色を選定し、絞りを決定し、遮光板パターン(gobo pattern)が選定される。マイクロプロセッサ200は図5に示すように制御およびデータバスを有し、制御およびデータバスはラッチ256およびタイマ258に接続してある。図5に示す割り込みラインおよび受信確認ラインは、さらに、割り込みエンコーダ回路260に接続されている。ラッチ256に保持されたデータは、複数のラインを介してPAL(programmable array logic)262に転送される。PAL262は、ケーブル264およびイネーブルラインを介して、パワー増幅器266に送られる制御コマンドの組合せを生成する。パワー増幅器266は、ライン群270を介してステッパモータ272に伝送される一連のパワー信号を生成する。ライン群270上のパワー信号により、ステッパモータ272は一連のステップで、所望の角度位置まで動く。

【0057】タイマ258はステッパモータ272のオペレーションに必要なタイミング信号を生成する。これらのタイミング信号は割り込みエンコーダ回路260およびPAL262の両方に供給される。よって、ステッパモータ272が位置を変える必要があるときは、ラッチ254にデータとして送られる制御コマンドがマイクロプロセッサ200により生成される。ラッチされたデータは、ついで、PAL262に転送される。PAL262はこのデータを制御信号に変換し、得られた制御信号はパワー増幅器266により増幅され、ステッパモータ272に供給される。ステッパモータ272の必要とする各オペレーションが行われたとき、適正な割り込みまたは受信確認のコマンドがエンコーダ回路260を介してマイクロプロセッサ200に伝送される。

【0058】パラメタ制御回路278を図7に示す。パラメタ制御回路278はポジション検知を必要とする機械的制御パラメタ装置と共に用いられる。本実施例では、パラメタ制御回路278は3つのホイールおよび1つの絞りを制御するのに用いられる。各ホイールおよび絞りはセンサを有する。ステッパモータにより色ホイールのようなホイールをオペレートさせる例を示す。このホイールはセンサ280により検知されるマークすなわち磁石を含む。センサ280はインデックス・センス回路282によりオペレートされる。検知されたインデックスは増幅器284の非反転入力端子に供給される。固定基準電圧は抵抗器286、288のオペレーションにより反転入力端子に供給される。増幅器284からの出力はバッファ290に供給される。バッファ290の出力は、各パラメタ回路のアドレス制御と、データと、割

り込みを、マイクロプロセッサ200に供給される。各割り込みの受信確認はバッファ290に供給される。

【0059】図8に、パンおよびチルトのようなパラメタに対し、ドライブおよびフィードバック制御するパラメタ制御回路296を示す。図4に示すように、マイクロプロセッサ200からのデータバスは、変更フィードバックのポジションおよび速度と、変更コマンドデータの速度との両方をサーボモータ298に供給する。速度制御データはラッチ300に入力される。ラッチ300により速度制御データがデジタル・アナログ変換器302に出力される。デジタル・アナログ変換器302はアナログ信号を生成し、そのアナログ信号はドライバ増幅器304の非反転端子に入力される。ドライバ増幅器304のドライブ端子はモータ298の端子に接続してある。回転計306はモータ298の速度を監視し、対応するアナログ信号をドライバ増幅器304の反転入力端子に供給する。フィードバック・ループが供給され、モータ298の回転速度を決定する。アナログ-デジタル変換器308に角速度情報がさらに伝送され、アナログ-デジタル変換器308により、デジタル速度情報がラッチ310に供給される。ラッチ310からの出力はデータ信号としてマイクロプロセッサ200に供給される。

【0060】モータ298はクラッチ312を介して直交位相エンコーダ314に物理的に連結してある。エンコーダ314からの2つの出力は、それぞれ増幅器316および318の第1の入力端子に供給される。各増幅器316および318の第2の入力端子は、電源およびグラウンドの間に接続した抵抗器により基準値に設定される。増幅器316および318からの出力はコンバータ320に供給される。コンバータ320はアナログ・ポジション信号をデジタル信号に変換し、このデジタル信号はクロック線および回転方向線を介してアップ/ダウン・カウンタ322に伝送される。アップ/ダウン・カウンタ322からの出力は、モータ298のポジションを示すものであり、データバスを介してマイクロプロセッサ200に伝送される。コンバータ320はさらに割り込み信号を生成し、かつ受信確認信号を受信するようになっており、これらの信号はマイクロプロセッサ200との間で交換される。

【0061】リピータ52は図2に示す各リピータと同様であり、図9でさらに詳しく述べる。リピータ52の目的は、ランプユニットとともにその他の制御されるステージ・デバイスと、制御コンソール24との間で、高速データ伝送を行うことにある。リピータ52はデータリンク26に直列に接続してある。リピータ52はバス38および40に対し両方向通信を行う。ランプユニットおよびコンソールは、それぞれ、ソース(source)およびデスティネーション(destination)と考えることができる。制御コンソールがソースであり、かつ、ランプユ

ニットがデスティネーションである場合についてリピータ52を説明する。

【0062】リピータ52は、好ましくは50Ω伝送ラインであるバス38、40を介して高速データを処理するように設計してある。リピータ52は、送信セクション332、すなわち図9に示す上部と、図9の下部に示す受信セクション334とを有する。

【0063】データリンク・バス38は変圧器336の入力端子に接続してある。抵抗器338および340は、それぞれ、バス38の2本の導線とグラウンドとの間に接続してある。さらに、データリンク・バス38は同じくグラウンドしたシールドを有する。変圧器336の2次側は増幅器342の非反転入力端子に接続してある。その反転入力端子はバイアス抵抗器345および344の間に接続してある。さらに、増幅器342の反転入力端子とグラウンドとの間にコンデンサ346が接続してある。

【0064】増幅器342の出力端子はManchesterエンコーダ回路352の入力端子に接続してある。Manchesterエンコーダ回路352からの出力は、インバータ354を介して、1つ以上の差動電流ライン・ドライバに送られる。インバータ354の出力端子は1つのこのようなライン・ドライバ356に接続してある。さらに、ライン・ドライバ356からの出力は、リピータ52のような別のリピータまたはランプユニットのような最終デスティネーションに伝送するバス38に接続してある。

【0065】受信セクション334では、バス40は変圧器358の1次側端子に接続してある。抵抗器360および362はバス40の導線およびグラウンドの間に接続してある。また、リンク40のシールドもグラウンドしてある。変圧器358の2次側は増幅器364の一方の入力端子に接続してある。増幅器364の第2の入力端子は抵抗器366と抵抗器368のノードに接続してある。コンデンサ370は抵抗器366と抵抗器368のノードとグラウンドとの間に接続してある。

【0066】増幅器364からの出力信号は、インバータ372を介してManchesterエンコーダ374の入力端子に渡される。さらに、Manchesterエンコーダ374の出力はインバータ376を介して差動ライン・ドライバ378の入力端子に渡される。ライン・ドライバ378からの出力は、データリンク26のバス40の差動端子を駆動するように接続してある。バス40は制御コンソール24に向けるか、あるいはリピータ52のような別のリピータの受信セクションに向けてある。

【0067】Manchesterエンコーダ352および374は、16MHzのクロック・レートの入力を提供する発振器382により駆動される。リピータ52は、さらに、抵抗器384およびコンデンサ386を直列接続したものを備えた始動回路を含む。この直列接続したものは正電圧源とグラウンドの間に接続してある。インバータ38

8はその入力端子が抵抗器384とコンデンサ386のノードに接続してある。インバータ388の出力端子はエンコーダ352および374のCTS入力端子に接続してある。インバータ388の出力端子はさらにインバータ390の入力端子に接続してある。インバータ390はその出力端子が各エンコーダ352および374のリセット入力端子に接続してある。パワーアップ時には、各エンコーダ352および374に入力されるリセット信号は、短期間の間、初期のロー(low)論理レベルである。コンデンサ386が充電されると、正規のオペレーションでは、リセット論理状態が変り、ハイ(high)論理レベルになる。従って、ディジタル回路、すなわち、Manchesterエンコーダは、初めに電源が投入されたとき、予め規定された状態にセットされる。

【0068】本発明に係る実施例では、Manchesterエンコーダ/デコーダ、例えば、エンコーダ352および374と、図4に示すエンコーダ168のようなものは、Harris Semiconductors Products Division Model HD-6409を備えている。Manchesterエンコーダ352および374は、モード選定入力端子に論理ハイレベルが結合されると、リピータ・モードが選定される。このManchesterエンコーダは高速データ・ストリームを受信するようにオペレートし、NRZ(nonreturn to zero)形式に変換される。クロック信号は慣用の方法で、データ・ストリームから回復される。ついで、データ・ストリームはインバータに出力される前に、タイミング再生(retiming)され、再構成される。このようにして、パルス幅と、進相または遅相のような歪も、データリンクでの伝送により悪化することがない。各リピータにおける高速データ・ストリームの再構成およびタイミング再生により、データリンク26によるデータ・エラー・レートが著しく減少する。

【0069】本発明の実施例の主な特徴は、各ランプユニットのオペレーションを分散制御することができることにある。この分散制御により、ハイレベルのコマンドがコンソール・プロセッサにより各ランプユニットにディスパッチされるようになっていく。これを「ブロードキャスト・コマンド」という。各ランプ・プロセッサは、その特定のランプ・プロセッサのプログラムと、前の条件により規定される適正な方式で応答する。これは、コンソール・プロセッサが各ランプユニットのステータスに関する現在の情報およびデータを全てストアするとともに、各ランプユニットの各パラメータをストアする従来のシステムと異なる点である。これらの従来のシステムでは、データ情報のキュー・ストリッジが全てコンソール・プロセッサ自体により完全に処理され、しかも、適正なランプユニットに伝送されたデータのみが、極めて詳細な命令、例えば、特定のステップモータを所望の角度だけ回転させるのに必要なパルス数のような命令であった。このことが次のように構成した本発明に係

るシステムと異なる点である。

【0070】本発明の実施例に係るシステムは、コンソールがその制御入力を読み取り、変化を検知したときに、この変化した入力を最小限処理し(例えば、スイッチの順番またはフェーダの名前を供給する)、この変化信号を、単一のハイレベル・メッセージで、全部のランプユニットに同時に伝送するように構成した。ついで、各ランプユニットはこの変化の所期の効果を認識し、ランプ自体のプロセッサで所望の応答を計算する。ハイレベルのコマンドを処理する際には、各ランプユニットのプロセッサは、他のランプユニットまたはコンソールとの対話を必要としない。例えば、コンソールのフェーダを動かしておくという単一のメッセージが、全て、ランプユニットに同時に伝送される。各ランプユニットのプロセッサは、キュー(cue)との個々の関連に基づき、リコール・キュー情報の平衡を再計算する。種々のランプユニットは1つのキューに対し異なるオペレーションを行い、ランプユニットの中には全くアクティブになることができないものもある。この新規な構成では、瞬時リコール用のキュー・メモリは全て個別の各ランプユニット・メモリに保持される。したがって、各ランプユニットはこの装置自体の中でキュー情報を全て利用できる。しかし、バックアップおよび長期保管、すなわち二次記憶のために、コンソール・プロセッサは各ランプユニットに対しキュー・データのコピーを保持する。このバックアップはディスク記憶装置に保持され、ランプ交換におけるシステム初期化時か、あるいは完全なメモリ切り替えのために、ランプユニットの各記憶装置に読み込まれる。

【0071】以上述べたことから明らかなように、大規模のキュー・データが、1回だけ、すなわち、システム初期化時に、狭帯域幅の通信リンクを介して伝送されるので、本システムの効率および信頼性を向上させることができた。その後、キューデータが各ランプユニット内で利用可能になる。各ランプユニット内では、キューデータの読み出し書き込みが高バンド幅のローカル記憶装置の環境で行われる。上述したことから明らかなように、本システムの効率は最適化され、特に、新たに発行されたコマンドに応答して、各ランプユニットのアクティビティに完全性が存在する状態で最適化される。コンソールからのコマンドは、システム全体として、各ランプユニットに、ブロードキャスト・コマンドとして1回の伝送で簡単に伝送される。各ランプユニットに必要なアクティビティは、他のランプユニットのアクティビティに関係なく、かつ、コンソールからさらにデータ伝送が行われなくても、行われる。よって、時間を著しく節約できるとともに、信頼性を著しく向上させることができる。これは、データリンクによる伝送の並列性に起因する。さらに、本システムにさらに多くのランプユニットを加えても、コンソール・プロセッサにも、データリ

ンクにも大幅な負担にはならない。本システムはランプユニットを追加したときにも、最適なステートに保たれる。というのは、各ランプユニットは必要な処理能力と、その機能を果たすのに必要な記憶装置とを加えるからである。1つのランプユニットが本システムに追加されても、コンソール・プロセッサの作業には極めてわずかしかな負担が加わらない。

【0072】上述したことに留意して、コンソールを本プロセッサ・システムの機能に関連して次に説明する。図10はコンソール・プロセッサ複合体の主要な機能をフローチャートで示す。コンソールを最初にパワーアップすると、コンソール回路は予め定めた内部変換により初期化され、プロセッサはメイン・シーケンサ・プログラムをコンソール回路に入力する。このプログラムは、予め定めかつ不変のシーケンスで、他の副プログラムに分岐する永久ループに似ている。各副プログラムがそのシーケンスで入力されると、このプログラムはメイン・シーケンサ・ループに戻る前に具体的な機能を果たす。

【0073】コンソールの副プログラムのうちの1つは、スイッチ入力センス・プログラムである。このプログラムにより、フロントパネル上のコンソール・スイッチが完全に走査される。任意のスイッチの押下または押下解除は、プロセッサ複合体によりセンスされる。そして、押下または押下解除がセンスされた各スイッチに対し、適正な応答ルーチンが活動化される。新たに活動化された各スイッチのステータスは、この応答ルーチンに転送される。

【0074】スイッチ入力センス応答ルーチンは、あるスイッチが押下されるかあるいは押下解除されたときに取られるアクションを指定する個別のスク립トである。スイッチの中には、機能的にグループ化され、従って、同じ応答ルーチンを採用しているものがある。この場合、共通のグループ内のスイッチの番号は、応答ルーチン間で識別される。この応答ルーチンでは、スイッチの番号は、グループ内の全てのスイッチに共通のスク립トで、スイッチ識別子として用いられる。次にその例を説明する。

【0075】コンソール・プロセッサが予め定めかつ不変のシーケンスでエンタする第2の副プログラムは、光学エンコーダ入力走査プログラムである。上述したように、種々のコンソール・デバイスの回転ポジションが決定され、その決定に従ってアクトする。コンソール・フロントパネルの回転入力デバイスは、慣用設計の光学エンコーダ/ハードウェア・カウンタ回路を備えている。光学エンコーダ入力走査プログラムは、各エンコーダに対するカウンタ値を読み出し、かつ、新たな値を、前回のスキャンに従ってストアされた値と比較するのに使用可能である。この比較により回転デバイスの位置の変化が示された場合、そのエンコーダに対する識別子が値の変化量と組み合わせられる。その結果が、コマンド・メッ

セージとしてネットワークを介して全てのランプユニットに送信される。ランプユニットは、コンソール・デバイスの回転ステータスの変化が特定のランプユニットで応答を必要とするか否かを個々に判定する。

【0076】フェーダ入力スキャン副ルーチンは、主シーケンサと出会う第3のルーチンとして現われる。このルーチンは、コンソール・パネルにあるスライダ・フェーダ制御デバイスのポジションの変化に応答する。フェーダは本質的には抵抗ポテンシオメータであり、その直線運動はアナログ・デジタル変換器によりセンスされる。このようにして、フェーダのポジションを変えると、新たにデジタル・コード化された数がセンス回路の出力端子に供給される。他のセンス回路を用いても同様の効果があることは当然である。フェーダ入力走査プログラムは、各フェーダ・センス回路の現在入力値を読み出し、この値が以前にストアした値から変わった場合にのみ応答する。光学エンコーダ入力走査プログラムの場合と同様に、フェーダのセンスにより新たなポジションを示した場合、フェーダ識別子はフェーダから読み出した実際の値と結合される。しかも、この情報はコマンド・メッセージの一部として全てのランプユニットにネットワークを介して送信される。ランプユニットは、それぞれ、フェーダ識別子およびランプユニットの内部ステートに基づき、新たなフェーダ値が適応可能か否かを判定する。

【0077】保留メッセージ管理副プログラムは、主シーケンサにより予め定めかつ不変のシーケンスで入力された追加プログラムを備えている。ある環境では、個々の応答ルーチンに従って対応するメッセージを伝送できるよりも速く、オペレータによりコンソール・スイッチを活動化させることができる。従って、応答ルーチンは、以前のメッセージがコンソール・プロセッサ複合体によりネットワークに伝送されていないことが分ると、各応答ルーチンにより、保留メッセージ・パケットが各応答ルーチンにより生成される。このパケットは、前のメッセージを完了し伝送されたとき、送信される。保留メッセージ管理副プログラムは任意の保留メッセージ・パケットが存在するか否かを確認するため、種々の副プログラムを走査し、かつ、関連する前のメッセージが伝送されたか否かを走査する。ついで、走査した結果、前のメッセージが完了している場合、保留メッセージ・パケットに対応するコマンド・メッセージが保留メッセージ管理副プログラムによりディスパッチされる。

【0078】キャラクタ・ディスプレイ制御副プログラムは主シーケンサにより入力され、コンソール・フロントパネルの英数字ディスプレイ・デバイスとして供される。これらのディスプレイは、幾つかのスイッチ入力応答ルーチンにより制御される。キャラクタ・ディスプレイ制御副プログラムは、応答ルーチンに対し共通の制御インタフェースを供給する。さらに、キャラクタ・ディ

スプレイ制御副プログラムは、ディスプレイ・データを、コンソール・システムに用いられるフォーマットから、英数字ディスプレイ・デバイスに対する一連のコマンドに変換する。

【0079】最後に、スイッチ・ランプ制御副プログラムが提供される。このプログラムは種々のスイッチとしてランプを制御し、スイッチが押下されたステートにあるか、あるいは押下解除されたステートにあるかをオペレータに指示する。このようにし、かつ従来のコンソール・スイッチ・システムとは対照的に、ランプ・パワーを搬送する電気スイッチ接点を必要としない。このことは、多くのコンソール・スイッチの信頼性を高めるのに実質的な効果がある。スイッチ・ランプ制御副プログラムにより送信されるランプ・オン・オフ・データは、応答ルーチンにより、コンソール・プロセッサ複合体の記憶装置に置かれる。また、スイッチ・ランプ制御副プログラムによるデータの検索と、最新の走査とを比較し、新たに押下されたスイッチに関連するランプを点灯するかあるいは消灯するかを判定する必要がある。

【0080】副プログラムを示す図10は、関係付けをしたプログラムを示すブロック図である。これらの関係付けをしたプログラムは副プログラムの種々のルーチンにより入力可能である。特に、これらの関係付けをしたプログラムは、コンソール電気装置によるあるハードウェア割り込みが生じたときに入力される。関係付けをしたプログラムは、それぞれ、種々のハードウェア機能か、データ構造か、コンソール論理ステートの態様を制御するルーチンを統合した集合である。このような関係付けをした1つのプログラムは、通信管理プログラムを備えている。通信管理プログラムの主要な機能は、コンソールおよび複数のランプユニットの間の伝送ネットワークを制御することである。種々の応答ルーチンによりデマンドされるネットワークに、データを順序よく伝送することは、各応答ルーチンによる要求の緊急度に応じて情報を順序よく流すことを保証するのに重要である。伝送ネットワークに並列特性があることは、1つのランプユニットに発生した障害が他のランプユニットの伝送能力に影響を及ぼさないかぎり極めて望ましい。これは、典型的に採用されている「デジーチェーン」すなわち直列接続のネットワークとは対照的である。上述したように、コンソールとランプユニットの間の通信パスは、全2重パス、すなわち、データを独立かつ同時に伝送することができる送受信パスである。通信管理プログラムはランプユニットとそのランプユニットにあるデータ送信器とを制御し、任意の時点で同時に、1つのランプユニットのみがネットワーク伝送パスを用いることを確認することができる。

【0081】通信管理プログラムに従って、2種類のメッセージ・アドレス、すなわち、各別のランプ・アドレスおよびブロードキャスト・アドレスが供給される。本

システムの各ランプユニットは、特定のランプユニットに関係付けをした一意のアドレスを伝送することにより、コンソール・プロセッサ複合体により個々にアクセス可能である。上述したようにネットワークに接続される各ランプユニットは、ランプ・アドレスを受信することになる。しかし、伝送されたアドレスのみが応答することになる。一方、ブロードキャスト・アドレスは、ネットワーク内の全てのランプユニットが応答する特定の値を有するランプ・アドレス・フィールドを含む。さらに、各ランプユニットは個々のランプ・アドレスに関係なくブロードキャスト・アドレスに応答する。

【0082】コンソールは2つの異なるカテゴリーのコマンド・メッセージに対し、ブロードキャスト・メッセージおよび個々のランプユニット・メッセージを利用する。個々のランプユニットへのメッセージのみを用いて、記憶ディスクにキューデータを保持し、かつ、各ランプユニットのステータスを報告し、かつ、ネットワークに新たに接続されたランプユニットに応答する。本システムの他の機能は全てブロードキャスト・メッセージにより果たされる。ブロードキャスト・メッセージは、例えば、ランプユニットに伝送され、ランプユニットをマニュアル制御するか、あるいはマニュアル制御から外す。変更コマンド・メッセージをブロードキャストし、かつ、ランプユニットを応答させることにより、ランプユニットがマニュアル制御される。さらに、キュー番号をブロードキャストし、かつ、各ランプユニットがこのキューを利用可能か否かを判定することができることにより、キュー情報データはコンソール・プロセッサ複合体により各ランプユニットから呼び出される。システム全体が一度初期化されると、演技の進行中に、各ランプユニットの必要とする機能は、全て、ブロードキャスト・メッセージの性質を帯びる。このアーキテクチャにより、ショーの演技は、コンソールにより連続的にデータが伝送される1個のランプユニットに発生する障害によつては損われないし、各ランプユニットからコンソールに向けた全2重ネットワークの半分がコンソールの専用となる。ネットワークの全2重伝送線のもう半分、すなわち、コンソールからランプユニットの間の部分は、コンソール情報をランプユニットに伝送する状態になっている。その結果、各ランプユニットはコンソール・スイッチと、調光器と、回転エンコーダ等のステータスの変化に反応することができる。特にランプユニットに伝送されたメッセージがランプユニットにより受信されたことは、ランプユニットからコンソールに伝送して受信確認される。ランプユニットからの応答が受信されない場合、通信管理プログラムはコマンド・メッセージを再送することになる。この再送により、雑音またはその他の問題に起因してランプユニットが誤伝送するという影響が打消される。

【0083】しかし、コンソール・プロセッサ複合体が

何回か再送しても、ランプユニットから応答がない場合は、このことはこのランプユニットがもはやオペレートしていない、ことを示していると、受け取られる。各ランプユニットにより伝送される選定したメッセージは、コンソールへのデータ伝送を含む。匹敵する方法では、第1の伝送に回答してコンソール・プロセッサの簡単な応答が受信されない場合は、データはランプユニット・プロセッサ複合体の通信管理プログラムにより再送される。より重大なネットワーク通信ラインの問題がある場合、コンソールはブロードキャスト・メッセージを多くとも3回伝送し、少なくとも1つのこのようなメッセージが雑音の多い通信ラインを介して受信されることを確認する。メッセージが伝送された回数に対応する順序番号がブロードキャスト・メッセージと共に伝送される。種々のランプユニットの通信管理プログラムは、順序番号を用いることにより、後続するコンソール伝送の繰り返しを無視する。制御複体内の通信管理プログラムは、種々のコンソール・プログラムに従ってコンソール・メッセージを受信し、ランプユニットに伝送する。このようなメッセージを待ち行列化する。特定のメッセージがランプユニットからの応答を必要とする場合、コンソール・プロセッサは応答を待ち、受信したとき、この応答をプログラムに戻し、次のメッセージの伝送に先立って、メッセージを初期化する。

【0084】関係付けをしたプログラムを示す図11は、ファイル管理プログラムを示す。ファイル管理プログラムは、ディスク・ファイル・システムを監視し、ランプユニット・キューデータを、順次、相対レコードとキーインデックス付きファイルとに供給する。各ランプユニットに関係付けをしたキューデータは、ランプユニットに割り当てられたコンソール制御チャネルを含むファイル識別子により識別される。プログラムされたコンソール・データも、各プログラマブル・コンソール機能に対し、1つずつ、ファイルによりディスクにストアされる。その他の全ての点で、ファイル管理プログラムは慣用の方式でオペレートする。図示の関係付けをしたプログラムもディスクデータ管理プログラムを含む。慣用の方法で、ディスクデータ管理プログラムは、ディスク内の自由セクタのリストを管理し、各セクタを種々のファイルに割り当て、ファイルの所望のセクタを突き止め、適正なアクションを実行するのに必要なディスク・ハードウェア信号を発行する各機能を供給する。このプログラムは、本発明の種々のインプリメントに採用される異なるディスク・ドライブを制御する修正を必要とする。この図には、別の関係付けをしたプログラムを例外表示管理プログラムとして示す。この例外表示管理プログラムはコンソールのフロントパネル上の英数字ディスプレイ・デバイスのうちの1つのデバイスのコマンドを使用し、オペレータの注意を英数字ディスプレイ・デバイスに引き付ける。これらの状態は、一般的に、コンソ

ールの操作中に生ずる。これらの状態では、問題を解決するのに必要な各オペレータの受信確認または援助を必要とする。英数字ディスプレイ・デバイスにディスプレイするためのディスプレイ・データのスク립トが、オペレータを援助するため、供給される。ディスプレイされたデータは予測され、かつ、活動化を必要とするスイッチ入力応答を含む。問題が解決されると、英数字ディスプレイ・デバイスの制御はキャラクタ・ディスプレイ・プログラムに返される。

【0085】ネットワーク・ステート制御プログラムは、各ランプユニットのネットワークに対するコネクションまたは切断の管理を保持する。ランプユニット・コネクションがまず通信管理プログラムにより検知されると、ネットワーク・ステート制御プログラムは信号を送信する。この場合、新たに接続したランプユニットから報告されるステータス・ビットに対して一連の検査が行われる。これらのビットはコンソールの前提条件である、ある条件およびアクションを表し、完全にオペレーション可能なランプユニットを認識する。これらの各ステータス・ビットに対して応答ルーチンが供給される。これらの応答ルーチンは現れた各ビットに基づいて、コンソールの操作を指定する。ネットワーク・ステート制御プログラムにより果たされる機能の中には、例えば、追加ランプユニット・プログラム・コードをダウンロードする機能と、ランプユニットに対するキュー・データをダウンロードする機能と、種々のコンソール・フロントパネルの調整つまみの現在のステートを記述するデータのバケットを伝送する機能がある。

【0086】ディスク・ステート管理プログラムは、ディスク・ドライブに対するディスクの挿入または取り出しを監視する。ディスクの挿入または取り出しに対して、コンソール・プロセッサ割り込みが生ずる。更新されたキュー情報をディスクに保持することは重要であるので、ディスク上でランプ・キューデータのコピーを更新するのを禁止する状態を通知することは、コンソールのオペレータにとって最も重要である。例外表示管理プログラムにより、このような誤動作通知を、コンソール・オペレータに気付かせている。このような状態はディスクの適正な組合せがディスク・ドライブに存在しないときに生じる。

【0087】本発明によれば、正規には、リアルタイム・クロック情報を各ランプユニットにブロードキャストするように稼動するネットワーク・リアルタイム・クロック・プログラムが供給される。リアルタイム・クロック情報は日付および時間の情報を備えている。このデータはコンソール回路の電池駆動の集積回路から生じ、通信管理プログラムによりランプユニットに送信される。このネットワーク・リアルタイム・クロック・プログラムはハードウェア割り込みにより活動化される。

【0088】正規の順序の演出またはショーの間に、コ

ンソールはこのコンソール上の各ランプから、ランプ・ステータス・データを正規に要求する。ステータス・ビット、例えば、キュー・データ・ダウンロード要求ビットのようなビットは、ネットワーク・ステート制御プログラムを活性化させる。他のビット、例えば、電球故障ビットのようなビットにより、上述したように、例外表示管理プログラムによりオペレータに通知される。さらに他のビットはコンソール・オペレータが後で試験するためにストアされるに過ぎない。ランプ・ステータス走査プログラムも、ハードウェア割り込みにより活性化される。割り込みに応答して、ランプのステータスが要求され検索される。割り込みを生ずるハードウェア・タイマは連続的にオペレートするので、コンソール・プロセッサ複合体は、通信ネットワークに接続された全てのランプユニットから最も新しいステータス情報を利用できる。

【0089】マルチプル制御装置ネットワークのオペレーションを図2および図11を再び参照して説明することができる。両方向バス80は制御コンソール24と、制御コンソール84と、代替制御コンソール82と、遠隔制御装置81との間にデータ通信を提供する。1つのインプリメントとして、バス80はデータリンク26と同様に電氣的に構成されており、制御コンソール24は図11の関係付けをしたプログラムで記載したような通信管理プログラムを有する。このプログラムは、通信管理プログラムがデータリンク26上のアクティビティを制御するのにサードするのと同様の機能であって、バス80上のアクティビティを制御する機能をサードする。2種類のメッセージ・アドレスが供給される。2種類のメッセージ・アドレス、すなわち、個々のコンソール・アドレスおよびシステム・アドレスは、図11の記述と同様の機能を与える。例えば、個々のコンソールには、メインコンソール・プロセッサ複合体が、特定のコンソール装置に関係付けをした一意のアドレスを伝送して個々にアクセス可能である。システム・アドレス・コマンドで、ネットワークに接続されたコンソールは全て応答することができる。

【0090】制御コンソール24によりシステム・アドレスに送信されるメッセージは、ランプユニットから受信されたステータス・データ・メッセージと、メイン制御コンソールの調整つまみのステートと、メイン制御コンソールにより処理されフォーマットされたシステム・ステータス・データとを含む情報を含む。これらのメッセージにより、制御コンソール24、82、84およびリモート制御装置81は、メイン制御コンソールと同様に表示することができるか、あるいは、異なる情報を表示することができる。制御コンソール24、82、84およびリモート制御装置81から主コンソールに送信されるメッセージには2つのタイプがある。一方のタイプのメッセージはメイン制御コンソールによりランプユニ

ットに送信されるメッセージと同一のフォーマットである。これらのメッセージには、これらのメッセージを発信したコンソールを識別するデータが含まれる。上述したように、ランプユニットに送信されたメッセージの中には、ランプユニットが応答するものもある。この応答も発信コンソールを識別するデータを含む。このデータにより、メイン制御コンソールはその応答を発信コンソールにルーティングする。

【0091】制御コンソール24、82、84およびリモート制御装置81からメイン・コンソールに送信されるメッセージのうちのもう一方のタイプのメッセージは、メイン制御コンソール自体に送信されるメッセージである。これらのメッセージの中には、メイン・コンソールのフロントパネル制御を操作するオペレータにより行われる1つのアクションまたは一連のアクションを複製するものもある。これらのメッセージにより、フロントパネル制御が物理的に操作された場合に、送信されるメッセージと同一のメッセージがメイン・コンソールによりランプユニットに送信される。他のメッセージにより、メイン・コンソールは、キューデータと、ランプユニットおよびメモリおよびメイン・コンソールのディスクにストアされているプログラムされたコンソールデータを修正する。

【0092】リモート制御装置の例としては、ハンドヘルド・デバイスがある。このデバイスは、照明デザイナーがステージに携帯し、これを用いてランプユニットの向き（アジマス）および高さ（エリベーション）を微調整し、あるセットまたはステージのある領域に光ビームを確実に当てるか、あるいは当てないようにする。制御コンソール82をディレクタ・コンソールにすることができる。そうした場合、照明ディレクタがリハーサル中に制御コンソール82を用いて、ランプユニットによりパフォーマンスされる現在のキュー以外のキューに対するデータを表示するか、あるいは、オペレータが制御コンソール24から離れたときに、ランプユニットのキューをリコールすることができる。

【0093】制御コンソール82は米国出願641,031号（発明の名称：Creating and Controlling Lighting Designs）に開示されたような制御装置である。この制御装置は予め記録されたコマンドと、手を用いないパフォーマンスの実行とを制御する。米国出願641,031号の番号をここに記載して実施例の一部とする。

【0094】制御コンソール82は米国出願693,366号（発明の名称：Improvement In High Intensity Light Projectors）に開示されている制御装置である。この制御装置により供給されるものにはビデオ・フォーマットのコマンドを含む。米国出願693,366号の番号をここに記載して実施例の一部とする。

【0095】制御コンソール84はある種の演技に対してステージをより適正に見ることができるポジションに

置かれている。制御コンソール84によりメイン制御コンソール24を動かす必要もなく、しかも、データリンク26に接続する必要がない。

【0096】制御資源ネットワークの他の例は、バス80と、バス80に接続された制御装置との通信では、例えば、マスタコンソール24のセッティング・ステートをストアし、しかも、あるモードのオペレーションの間、マスタコンソール24のセッティングまたは修正をリコールするかあるいは「プレイ」する、記憶機能およびプレイバック機能を有するユニットである。

【0097】両方向バス80の他のインプリメントが可能であり、そのインプリメントには、LAN(local area network)が含まれ、制御コンソール24と単一の制御コンソール84の間のポイント・ツー・ポイント・データ・リンクが含まれる。その上、追加または代替制御コンソールか、あるいは、追加または代替リモート制御ユニットを、特注のコンソールではなく、汎用コンピュータにインプリメントすることができる。

【0098】以上、コンソール・プロセッサに利用可能な種々のプログラムを説明した。次の例は、オペレータによりある「チャンネル選択」コンソール・ボタンが押下されたのに応答して、種々の前記プログラムを実行する例である。このボタンを押下すると、あるランプユニットがマニュアル制御可能状態になる。さらに別のコンソールつまみを回すと、ランプがその複数の軸のうちの1つの軸のまわりを回転させられる。次の説明で、コンソール機能を各ランプユニットに移すという非集中制御の効果が理解されるであろう。また、本発明により、通常のプロセッサ制御照明システムに比べて、コンソールの必要とする処理が著しく減少する。コンソールおよびランプユニットでタスク共用することにより、システム・パラメタを変える速度が速くなる。さらに、開示した実施例では、コンソールは、もはや、本システム内の各ランプユニットに対し多量のデータを順次処理する必要はない。その代りに、ランプユニット・プロセッサは、それぞれ、その装置に対して変更する必要があるアクションを完遂する。さらに、本システムの場合は、単一のランプユニットが必要とするとき、システム全体を変更することができる。メッセージを全てのランプユニットに同時に伝送するので、本システムにランプユニットを加えても、伝送速度が比例して遅くなる（このことは従来のシステムでは典型的なことであった）こともない。マニュアル操作によりステージ・ランプのポジションを変更する例では、制御コンソールが通常の初期化ルーチンをパフォーマンスするものと仮定する。また、コンソール・プロセッサがランプユニットの通信を確立し、各ランプユニットに、個々の初期化に必要とするデータを全て供給し、そのシステムが、メインシーケンサの永久ループでオペレートするものと、仮定する。この永久ループでは、メインシーケンサは、オペレータからコンソール・

デバイスを経由した入力待。逐次化ルーチンの間、スイッチ入力センシング・プログラムをコールする。スイッチ入力センシング・プログラムはコンソールのスイッチ入力ハードウェアを走査して、コンソール・フロント・パネル上のスイッチ表示(appearing)のマップを作成する。このマップでは、セット・ビットは押下されているスイッチを表し、クリア・ビットは押下されていないプッシュ・ボタンを表す。

【0099】このマップは、各スイッチのステータスを含むメモリのマップであって、前の走査で読み出されたようなマップと同様のマップのコピーと比較される。第1のマップと第2のマップを比較して、第1および第2のマップを生成する間に変更されたステータスを有するスイッチを示す第3のマップが作成される。比較した結果、変更がない場合は、そのプログラムはメイン・シーケンサに返される。変更があったとすると、そのプログラムは第3のマップを、ビットごとに走査して、変更されたスイッチを識別し、対応する応答ルーチンを活動化する。識別子と、新たに活動化されたスイッチの新たなステータスとを、関係付けをした応答ルーチンに渡す。新たにオペレートされたスイッチは、「チャンネル選択」スイッチ・グループであって、同一の応答ルーチンによりサービスされるスイッチ・グループのメンバとして識別される。スイッチ識別子は「チャンネル選択」スイッチ・グループ内のスイッチの数を示し、スイッチ識別子は、追加のグループ・セレクトとともに、スイッチが対応する制御チャンネルを識別する。1000個のコンソール制御チャンネルは、それぞれ、コンソール・メモリ・マップの単一ビットにより表され、単一ビットはチャンネルがマニュアル操作のために選択されたか否かを示す。スイッチが押下されている場合は、そのチャンネルに対するビットの値は反転されており、マニュアル操作のためのランプが選択される。そのランプが既にマニュアル操作されているときは、そのスイッチを押下すると、ユニットのランプのマニュアル操作が解除される。既に説明したマップでは、1ビットのみが変更されるが、マップ全体が全てのランプに同時にブロードキャストされる。各ランプはそのマップを試験し、ブロードキャストされたメッセージに基づき、その制御が変更されたか否かを判定する。ネットワーク全体にこのマップを伝送した後は、そのスイッチの押下に応答して、コンソールを処理する必要はない。

【0100】スイッチの押下に応答して応答ルーチンに入り、ブロードキャスト・タイプのメッセージを送信するコマンドで、通信管理プログラムをコールする。このブロードキャスト・タイプのメッセージはそのメッセージ・データを保持するメモリ・ブロックを指すポイントを含む。その通信管理プログラムは、通信機能をインプリメントしたプログラマブル集積回路を開始するか、あるいは既に通信中である場合、通信管理プログラムは、

現在のメッセージ伝送を終えた後、次の伝送のためにコマンドおよびメモリ・ポインタを待ち行列化する。通信処理に必要な任意の追加処理は、種々のプログラマブル集積回路からのコンソール・プロセッサ割り込みの応答としてパフォームされる。スイッチ活動化と協働して、通信管理プログラムをそれ以上処理する必要はない。

【0101】通信管理プログラムによるメッセージの伝送を完遂するか、あるいは、将来、伝送するため、メッセージを待ち行列化するとき、この通信管理プログラムは、応答ルーチン・プログラムおよびスイッチ入力センス・プログラムにより、メインシーケンサに戻る。その結果、メインシーケンサは、新たに押下されたスイッチをセンスしたとき、前回終了したポジションで永久ループに入る。このメインシーケンサは、マニュアル制御のために押下されたランプ・ボタンの押下が解除されるまで、続行する。再び、スイッチ入力センス・プログラムに入り、そして、各走査マップを比較して、スイッチ・ステートの変化が示される。スイッチは、上述したように、再び識別され、関係付けをした応答ルーチンが活動化される。

【0102】この応答ルーチンはスイッチが押下解除されたときにもアクションがない。このことは、スイッチが押下されるか、押下解除されたときに、応答ルーチンを活動化する他の種類のスイッチとは対照的である。いずれにしても、スイッチ入力センス・プログラムにより応答ルーチンからメインシーケンサに戻される。再び、メインシーケンサはその永久ループ内で走査をレジュームする。このループを抜け、光学エンコーダ入力走査プログラムに入る。人間のオペレータが適正なコンソール・デバイスを回転させることにより、適正な舞台ランプが対応して回転する。エンコーダ/カウンタ回路は、上述したように、光学エンコーダ入力走査プログラムに数値を入力する。各エンコーダ/カウンタ回路により生成された値は、エンコーダ軸がオペレータにより回転されると変化する。スイッチ入力センス・プログラムと同様に、光学エンコーダ入力走査プログラムは、走査ごとに読み出した値を、前回の走査に関して記憶した値と比較する。比較した結果、差がある場合には、適正なランプ・コマンドが生成される。メッセージ・ブロックは、マニュアル変更ランプ・コマンドと、変化量と、特定のエンコーダに対する識別子とを含む。ついで、ランプコマンドは通信管理プログラムにブロードキャスト・メッセージとしてディスパッチされる。全てのランプ装置はブロードキャスト・メッセージを受信し、このメッセージの特定のランプ装置に対して、適用可否かを判定する。

【0103】既に述べたように、通信管理プログラムはこのメッセージを直ちに伝送して処理するか、あるいは通信チャンネルがクリアのとき、このメッセージを次の伝送のために待ち行列化する。ついで、コンソール・プロ

グラムはメインシーケンサの永久ループに戻る。上述したことは、オペレータにより指定されたように、コンソール・プロセッサによるランプ位置変更に、コンソール・プロセッサが参加することである。追加処理および後処理は、全て、必要な場合、個々のランプ・ユニットにより行われる。

【0104】本発明の原理を説明する次の例は、特定のランプ・ユニット・プロセッサ記憶装置にキュー・データ情報をストアする例である。この機能は、「ストア・キュー」スイッチを押下すると、コンソール・オペレータにより開始される。上述した例に関しては、メインシーケンサが永久ループを抜け、スイッチ入力センス・プログラムに入る。このスイッチ入力センス・プログラムは、新たに入力マップを読み取り、これを前回のマップにストアされた本システムのステータスと比較する。従って、「ストア・キュー」スイッチの状態は、押下されていることが分る。ついで、このスイッチが識別され、各応答ルーチンがコールされる。

【0105】「ストア・キュー」スイッチに対して適正な応答ルーチンは、2つの必要な条件、すなわち、「ストア・イネーブル」スイッチもまた現在押下されていることと、キュー番号が「ストア・キュー」スイッチの上方の表示窓に出力されていることとを検査する。これらの2つの条件が満足された場合、コンソールはストア・キュー・コマンド・ブロードキャスト・メッセージをネットワークを介して各ランプユニットに送信する。さらに、「ストア・キュー」ボタンの上方の窓内に出力されるキュー番号もまた同じメッセージに入れてブロードキャストされる。

【0106】通信管理プログラムは、全てのランプユニットにより同時に受信されるブロードキャスト・メッセージをネットワークを介してデータ伝送する。メッセージが伝送されるか、あるいは伝送のために待ち行列化された後、通信管理プログラムは応答ルーチンおよびスイッチ入力センス・プログラムによりメインシーケンサの永久ループに戻る。メインシーケンサは、必要に応じて、他のオペレータのコマンドを規定通りサービスする。しかし、この例のサービスとともに、これらのサービスの間にある他の多くのサービスでは、メインシーケンサは、ほぼ即時のアテンション(attention)を必要とし、かつ、規則正しくタイミングをとった割り込みにより、周期的に優先使用(preempt)される。規則正しくタイミングをとった割り込みは、コンソール・プロセッサに割り込みハードウェア・タイマ集積回路の周期的な割り込みによりランプ・ステータス走査プログラムを活動化するようになっている。ハードウェア・タイマにより割り込みが生じると、プログラムは異なるランプユニットに指令して、このランプユニットの現在のステータスを記述したデータを含むメッセージを、コンソールに送信する。このメッセージに現われるデータの種類の

は、ランプ・プロセッサ・システムに関して、後程説明する。

【0107】上述したように、ストア・キュー・コマンド・ブロードキャストのため、このシステムのランプユニットの中には、新たなキューデータ発生を報告を開始し、ディスクに記憶するためにコンソールに送信するものがある。ランプ・ステータス走査プログラムはシステム内の全てのランプを交互に処理し、新たにストアされたキューに含まれる全てのランプは、そのキュー・データをコンソールに最終的に送信することができる。ランプ・ステータス走査プログラムは、ステータス・リード(read)コマンド・メッセージを通信管理プログラムによりランプユニットに送信して、個々のランプユニットに対するステータス・データを獲得する。

【0108】ステータス・リード・コマンド・メッセージは、ブロードキャスト・メッセージについて上述した方法とほぼ同様にして、通信管理プログラムにより個々にアドレッシングされる。しかし、ステータス・リード・メッセージ・コマンドは、特定のランプユニットからの応答を必要とするので、通信管理プログラムは、ランプ・コマンド・メッセージを伝送した後に、通信ネットワーク・チャネルをオープンした状態に保持する。この通信ネットワーク・チャネルは、ランプが応答するか、あるいは応答がないまま、ある期間が経過するまで、オープン状態に保持される。この場合、ランプに障害が生じたものと仮定する。さらに、ランプ・ステータス・プログラムでの処理は、ランプユニットから応答を受信するまで、一時停止状態に保持する。

【0109】一度、特定のランプユニットがステータス・リード・メッセージに応答すると、通信管理プログラムは受信したメッセージによりランプ・ステータス走査プログラムに戻る。この例では、受信メッセージ中のビットの1つは、ランプがランプユニット・プロセッサ記憶装置内にキューデータをストアしていることを示すことになる。このキューデータはディスク記憶のために、まだ、コンソールに転送されていない。コンソールの多くの入力走査プログラムと同様に、ランプ・ステータス・プログラムは入力値の変化にだけ反応する。ランプ・ステータス・データ内にセット・ビットが現われるとネットワーク・ステート制御プログラムを活動化する。この変化により、ランプ・ステータスの変化に応答する。ネットワーク・ステート制御プログラムは、ランプユニットから受信されたステータス・ビットを処理する応答ルーチン群を有する。これらの応答ルーチンの中には、ランプ問題、例えば、電球故障のような問題の通知を、コンソール・オペレータに供給するものがある。ネットワーク・ステート制御プログラムの他の応答ルーチンは、要求時に、各ランプユニットにプログラム・コードをダウンロードする。この例で受信されたデータビットに関連する応答ルーチンは、ランプユニットからのキュー

ーデータをアップロードしてこのデータをディスク・ファイル・システムの適正なファイルに記憶する。ネットワーク・ステート制御プログラムは、まず、コンソール・プログラム・ディスク・ステート管理プログラムに位置するフラグを検査し、ランプユニットからの新たなキュー・データを実際ストアすることができることを保証する。實際上、ディスクをキュー記憶のために利用可能な場合、応答ルーチンは、キュー・バッファ・アップロード・メッセージで、通信管理プログラムをコールし、同様に、そのデータがストアされるメモリの使用されていないセクションを指すポインタをコールする。キュー・データの記憶のためにディスクが利用可能でない場合、新たなデータはランプユニットからアップロードされない。その代り、コンソール・フロントパネル表示器が点灯され、キュー記憶装置をランプユニットからコンソールにアップロードする必要があることを、オペレータに気付かせる。このことをオペレータ・コマンドにより後で達成させることができる。

【0110】上述したステータス・リード・メッセージとほぼ同様に、キュー・アップロード・コマンドは特定のランプユニットに送信される。また、キュー・アップロード・コマンドにより、通信管理プログラムはランプユニット応答を得るため待機する。本発明の好ましい形態では、プログラマブル通信回路がセットアップされ、ネットワーク・ステート応答ルーチンにより指定されたメモリ空間に、ランプユニット応答をストアする。ランプユニットからコンソールへのデータの転送が完了すると、通信回路がコンソール・プロセッサに割り込む。通信管理プログラムが再び活動化される。従って、通信管理プログラムは通信伝送が完了し別のメッセージが保留である場合はこのようなメッセージの伝送を開始し、ネットワーク・ステート制御応答ルーチンに戻る。

【0111】ネットワーク・ステート制御応答ルーチンを含むことにより、ランプユニットから受信されたデータがファイル・レコードに細分割される。ファイルをレコードに細分割するのに用いたのと同じフォーマットがランプ・キュー記憶に用いられ、同様にディスク・ファイル・システムに用いられる。受信されたデータが幾つかのキュー・データである場合がある。というのは、ランプ・ステータスを走査する速度を、オペレータがキューを記憶する速度より一時的に遅くすることができるからである。この例では、キュー記憶コマンドからのデータのみが、操作されるデータであると仮定する。ディスク・ファイルは記憶されるキュー・データの前に存在したので、ランプ・キュー・データを既に含んでいる。そのため、必要なことは、ディスク・ファイルに適正なレコードを追加するか、あるいは再書き込みするだけである。応答ルーチンはこのことを次のようにして行う。すなわち、ファイル管理プログラムを呼び出し、キュー・データ・ファイル・ディレクトリ内の特定のランプ制御

チャンネル番号によりファイルを開いて行う。ファイル管理プログラムを呼び出して、キュー・データ・ファイル・ディレクトリ内のランプ制御チャンネル番号によりファイルを開く際に、応答ルーチンによってこのことが完遂できる。ついで、応答ルーチンはランプユニットから受信されるレコード・データを用いて書き込みコマンドをファイル管理プログラムに発行する。一度、このデータの書き込みが完遂すると、応答ルーチンがファイル管理プログラムをコールし、このコールによりファイルが閉じられる。

【0112】ファイル管理プログラムは、上述したように、キュー・アップロード応答ルーチンに対し3つの機能を果たす。キュー・データ・ファイルを開くコマンドにより、キュー・データ・ファイル記述子のディレクトリのファイル記述子が探索される。ファイル記述子が見付け出されると、その記述子を用いてファイルの第1のフラグメントを見付け出し、これをディスクからロードする。書き込まれるファイル・レコードは2つの部分、すなわち、キュー番号およびランプ機能データを備えている。キュー番号はレコードに対して一意のインデックスとして利用される。新たに受信されたレコードをファイルに書き込むコマンドが発行されると、メモリ内に既に現われているフラグメントをファイル管理プログラムが探索し、書き込まれているレコードのインデックスを見付け出す。このインデックスが第1のフラグメント内で見付け出せない場合は、ファイルの他のフラグメントが順次試験される。現在のレコードがファイルに書き込まれているレコードのキュー番号を既に含んでいる場合は、これを新たなレコードのランプ機能データでオーバーライトされる。ファイル内にインデックスが見付け出せない場合は、このファイルはレコードに追加される。ネットワーク・ステート制御応答ルーチンからのコマンドであって、ファイルを閉じるコマンドにより、ファイル管理プログラムはファイルに関係付けたメモリ内のデータを指すポインタを解放する。このようにして、ネットワーク・ステート制御応答ルーチンは、必要なときは常に、これらのメモリ空間を再使用することができる。そのファイルにさらにアクセスするには、ファイル・オープン・コマンドを発行しなければならない。ファイル管理プログラムがディスクに記憶されたデータをアクセスする必要があるときは常に、ディスク・データ管理プログラムが活動化される。このプログラムは、実際にディスク・コマンドを発行するディスク・ドライブ制御回路を制御し、ディスクからデータを読み取る。ディスク・データ管理プログラムは、現在使われているディスクの部分のアカウントを保持し、ファイル管理プログラムにより要求される特定のファイル・フラグメントをアクセスするのに必要なアクションを判定する。

【0113】図12および図13を説明する。本システム全体か、あるいは、ランプユニットにパワーが供給さ

れると、ランプ・ステート初期化プログラムが活動化される。このプログラムも、ある割り込みが生じてランプ・システムの主要な誤動作を指示したとき、正常なランプ・コンピュータ・オペレーション中に活動化される。さらに、特定のランプユニットの通信アドレスが変更された場合、ランプ・ステート初期化プログラムの一部が再エンタされる。ランプユニットは、それぞれ、種々の機能を果たすROMベースのプログラムを含む。例えば、ROMベースのプログラムはランプ・システムの適正なオペレーションに必要なあるハードウェアを試験し、このプログラムは装置内の種々のプログラマブル回路を、事前定義した公知のステートにプリセットする。さらに、このプログラムは、ランプユニットのある部分の検査を要求するスクリプトにより続けられ、試験の結果に従ってとられるアクションを前もって記述する。スクリプトの終りに、ランプユニットはコンソールと完全な同期状態になり、そして、プロセッサは自己検査と、物理状態監視と、コンソール・コマンド伝送に対する応答とによりなる永久ループに入る。

【0114】初期化プログラムにより行われる最初のタスクは、EPROM記憶装置からのプログラムが有効か否かのチェックサム・テストである。予めタイミングをとったソフトウェア・ループに対するハードウェア・タイマの試験も行われる。さらに、通信ハードウェアのループバック試験と、RAM記憶装置の一部の読み出し／書き込み試験も行われる。試験されたランプユニット回路が故障していることが分ると、実行を停止する。一度、ランプユニット・ハードウェアのオペレーションが試験されると、種々のプログラム・サブルーチンが実行され、プログラム変数を初期化し、通信に用いられるプログラマブル回路をセットアップする。各ランプユニットのID(identity)は適正な入力デバイスから読み出された通信アドレスである。本発明の好ましい形態では、各ランプユニットの識別は、3桁のつまみホイール・スイッチのセッティングにより行われる。

【0115】よって、1000個ものランプユニットが本システムに接続され、それぞれ独立のIDを保持する。ランプ複合体プロセッサの入力は、ランプ・ハードウェアに関係付けをしたサーボモータおよびステッパモータの構成を要する。ランプ複合体のアクチュエータの異なる組合せを制御するため、同じコンピュータ・ハードウェアおよび基本プログラムが利用されるので、ランプ・システム・プログラムの一部がランプ複合体で異なることになる。必要な場合は、特定のランプ複合体用の適正なシステム・プログラムを、コンソールからダウンロードすることができる。しかし、各プログラムが各ランプユニット内の書き込み可能な不揮発性メモリに保持されるので、これらの追加プログラムを各ランプユニットにダウンロードする必要はない。上述したように、このメモリは、バッテリーによりRAMメモリをバックアッ

ブして不揮発にされる。

【0116】次に、ランプユニットのRAMメモリに存在するプログラムを検査し、そのプログラムが有効か否かを判定する。チェックサム・テストが行われ、プログラム中の識別子を上述したアクチュエータ構成入力と照合する。その結果、プログラムが有効である場合には、内部フラグがクリアされ、よって、追加プログラムを実行させることができる。追加プログラムが無効である場合は、フラグがメモリ・ステータス・ワードにセットされ、制御コンソールはプログラムをダウンロードし、ランプ・システム・プログラム・メモリと置換する。これらの追加プログラムの実行をディスエーブルするフラグもセットされる。

【0117】このとき、通信管理プログラムが活動化され、コンソール・プロセッサ複合体とのコンタクトを確立させる。その後、コンソールがランプユニットの通信アドレスを問合わせると、通信管理プログラムが応答することになる。ランプユニットの特定の構成と、上述した有効性検査の結果がコンソール・コマンドに回答して報告される。これは、コンソール・プロセッサ複合体と、ランプユニットの複合体との間の初期通信の1つを構成する。

【0118】上述した検査の結果、ランプユニットRAMメモリの追加プログラムが無効である場合、各プログラムが制御コンソールからダウンロードされるまで、初期化の実行が延期される。ランプユニット・プロセッサは自己検査およびコンソール・コマンド応答の永久ループに入る。プログラム・ダウンロードに関係付けをしたコマンド応答ルーチン・プログラムの終りで、RAMメモリの追加プログラムの実行をディスエーブルするように、前回にセットしたフラグがクリアされる。ついで、ランプ・ステート初期化スクリプトが再び入力される。結局、これらの有効な追加プログラムの集合が、各ランプユニットのRAMメモリに存在することになる。ついで、追加プログラムに関係付けをしたサブルーチンが実行され、物理アクチュエータの制御のために用いる追加プログラム変数およびプログラマブル回路を初期化する。また、ランプユニット・プロセッサを割り込みに案内するアドレスのテーブルが修正され、追加プログラム内の割り込み応答ルーチンの存在を反映する。ついで、さらに多くのサブルーチンがコールされ、物理アクチュエータおよびフィードバック・センサの較正およびインデックス付けを機能させる。これらのサブルーチンにより、種々のアクチュエータをその全運動範囲に亘って動かし、任意のセンサのポジションにマーク付けをし、種々のアクチュエータおよびフィードバック・センサのオペレーションが適正か否かを検査する。

【0119】ランプユニットに対する通信アドレスを、このランプのオペレーション中に変化させる場合には、新たなアドレスに従って、コンソールとの通信を再び確

立する。ランプ・ステート初期化スクリプトを再入力させ、ランプユニットが新しいアドレスに対しコンソールと再び同期することを許可する。

【0120】ランプ・ステータス・ワード中のフラグはこのときにセットされ、コンソール・プロセッサ複合体をプロンプトし、コンソールのステートに関する情報を含むデータのバケットを送送する。このデータ・バケットはランプユニットにとって必要なものであり、ランプユニットは後のコンソール・コマンドに適正に応答させることができる。バケット内のデータの性質は、コンソールのあるサブセクションにおける制御の位置と、特定のランプユニットに割り当てたコンソール制御チャンネル番号とに関する情報を備えている。光度論制御装置にフラグをセットして、適正なデータがコンソールから受信されるまでは、特定のランプが点灯されないようにする。ついで、初期化プログラムは、コンソール・ステート・バケットを受信するまで、自己検査/コマンド応答ループに再入力される。

【0121】コンソール・ステート・バケットが完了すると、コマンド応答ルーチンと、ランプ・ステート初期化スクリプトが再び活動化される。コンソールから受信されたステート・バケットに関係付けをしたデータは、一時的に記憶され、一方、キュー・データ・メモリが有効か否かの追加の検査を行う。チェックサム・テストを行い、キュー・データ内の制御チャンネル識別子と、コンソールから受信された制御チャンネル識別子とが一致したか否かを試験する。その結果、チェックサム/チャンネル番号試験の間に、キュー・データが有効である場合は、ランプユニット・キュー・データに対する最後の更新時間の表記と、制御コンソール内のディスクに記憶されたデータの表記とを比較する。これらの更新時間が一致した場合は、処理を継続する。比較した結果、より新しいデータがランプユニット・メモリに記憶されている場合は、コンソール・オペレータ・アービトレーションが呼び出され、どのキュー・データを使用するかを判定する。ランプユニット・メモリに存在するデータよりも新しいデータがディスクに存在するか、あるいはキュー・データが無効である場合、ランプ・ステータス・ワードにフラグがセットされる。このフラグはコンソール・プロセッサをプロンプトして適正なキュー・データをランプユニット・メモリにダウンロードする。ついで、リwind・コマンドをキュー・データ管理プログラムに送信し、メモリ内のデータを消去し、自己検査/コマンド応答ループに再エンタする。

【0122】あるいはまた、有効キュー・データがランプユニット・メモリに存在した場合は、初期化スクリプトが再エンタされる。ついで、キュー・データおよびコンソール・ステート・バケットを利用し、全ての機能論制御部をセットアップし、次のマニュアル制御またはコンソールからのキュー・リコール・コマンドに回答す

る。キュー・リコール・コマンドが受信されると、光度論理制御プログラムのフラグがクリアされる。このフラグはリコールされるが、完全に同期していない光源の照度を抑制する。ついで、ランプユニットの正常なオペレーションを開始させることができる。この最終プログラムをセットアップした後、初期化スクリプトが終了され、上述した活動化基準の1つが生じるまで、シーケンサ・ループ内で引き続き処理される。

【0123】以上、一般的に、コンソールおよびランプユニット・プロセッサを交換して各ランプユニットを適正に初期化することを説明した。初期化後に、各ランプユニットにより行われるメイン・バックグラウンド・アクティビティは、個々のメインシーケンサ・ループ・プログラムによるものである。次に、図14を説明する。一般的に、メインシーケンサ・ループ内のランプユニット・プロセッサのアクティビティは、コンソール・プロセッサから受信される通信に対し入力バッファを走査することを含む。チェックサムの完全性を得るため、RAMメモリ内のキュー・データおよびプログラムを検査する。ランプユニットプロセッサはランプユニットに関係付けをした通信アドレスの変更も走査する。メインシーケンサ・ループは、コンソール・コマンド通信が受信されるか、あるいは、チェックサムに障害があるか、あるいは、アドレスが変更されるまで（メインシーケンサ・ループが一時的に終了される）、各ランプユニットで連続的に実行されるプログラムである。さらに、メインシーケンサ・ループ内の処理は、割り込みベースのアクチュエータ制御プログラムが活動化されるか、あるいは物理フィードバック割り込みが生じたとき、一時的に停止される。

【0124】メインシーケンサ・ループ・プログラム自体は、種々のサブプログラムの活動に対して永久に繰り返されプリセットされたサイクルである。これらのサブプログラムは、次に詳しく述べるが、コマンド・インタプリタと、メモリ・チェックサム検査と、通信アドレス走査サブプログラムを含む。各場合に、メインシーケンサ・ループがサブプログラムに入ると、検査を行う。この場合、メインシーケンサ・ループに再エンタするか、あるいは、行われる検査の結果に基づいて、応答が行われる。

【0125】コマンド・インタプリタ・サブプログラムに関して、永久ループ型のプログラムを活動化する。この場合、一連の命令が実行される。コマンド・インタプリタ・サブプログラムで実行される第1の命令またはアクションは、リード・コマンドを通信管理プログラムに発行することである。リードコマンドを発行した後に、メインシーケンサ・ループに戻る。コマンド・インタプリタ・サブプログラムがその後活動化されたとき、前回発行されたリード・コマンドのステータスが通信管理プログラムにより検査され、その検査の結果、リード・コ

マンドによる処理が完了していない場合は、メインシーケンサ・ループに戻る。リード・コマンドによる処理が完了したことが示されると、すなわち、検査ステータスがコンソール・プロセッサからの通信が完了したことを表すとき、コマンド・インタプリタ・サブプログラムは、コンソール・コマンド・メッセージにより発行され、かつ、新たに受信されたデータの第1のワードを試験する。コンソール・コマンドはランプユニット・プロセッサにより実行される。コンソール・コマンドがさらにコンソールからのデータ伝送を必要としない種別のものである場合、受信されたデータを一時的にストアし、コンソールにより送信される次のコマンドを取り出す別のリード・コマンドが発行される。受信され、かつ、ランプユニットROMメモリにストアされた関係付けをした応答ルーチンを有するコンソール・コマンドが、即時に実行される。RAMメモリ内に位置指定された追加プログラムが有効か否かが、他のコンソール・コマンドが実行される前に確認される。とにかく、コマンドが完了するか、あるいはコマンドが全て割り込みに基づいてさらに処理されるまで、コマンド応答ルーチン内で引き続き処理される。この場合、制御はメインシーケンサ・ループに戻される。特定の種別のコンソール・コマンドと、その関係付けをした応答ルーチンを次に述べる。

【0126】メモリ・チェックサム・サブプログラムが活動化されると、そのプログラムはプログラム・コードおよびキュー・データをストアしたメモリ・セクションの完全性を確認する。有効と確信されたメモリのこれらのセクションのみが検査される。プログラム・コードのチェックサム検査が失敗した場合は、適正なフラグがランプ・ステータス・ワードにセットされ、コンソールをプロンプトしてプログラム・コードをダウンロードする。さらに、コマンド・インタプリタ・プログラムのオペレーションは、プログラム・コードが置換され再び検証されるまで制限される。コンソールがプログラム・コードの必要なダウンロードに応答するときは、ランプ・ステート初期化スクリプトが上述したように再エンタされる。キュー・データが無効である場合、適正フラグがランプ・ステータス・ワードにセットされる。また、この場合、コンソール・プロセッサがプロンプトされ、キュー・データをダウンロードする。巻戻しコマンドはキュー・データ管理プログラムにディスパッチされ、無効なキュー・データをクリアする。有効なキュー・データをダウンロードした後は、処理は必要でない。両方の状態では、適正なアクションが行われると、制御はメインシーケンサ・ループに戻される。

【0127】通信アドレス走査プログラムとして識別されたサブプログラムは、ランプユニットの識別コードを読み取る。上述したように、識別コードは初めにセットされたディジット・スイッチにより設定され、通信ネットワーク内のランプユニットに一意のアドレスを供給す

る。このサブプログラムは、スイッチから読み取られた値をメモリ内のコピーと比較する。比較した結果、識別アドレスが変化している場合は、タイマを始動させる。このタイマはある期間の後に、ランプユニット・プロセッサへの割り込みを生ずる。走査中に読み取られた新たな識別アドレスは、メモリにストアされ、後の識別変化と比較される。新たな識別アドレスが検知されるごとに、タイマが再始動される。通信アドレスを変えると、タイマ割り込みが生ずるまで、他の応答は必要がない。スイッチ・デバイスのアドレス変更完了を確認するのに、例えば、5秒の時間間隔が好ましい。タイマ割り込みが生じると、ランプ・ステート初期化スクリプトが再エンタされる。アドレス変更が、注目したスクリプトに従い、かつ、上述したように処理される。

【0128】上述したように、コマンド・インタプリタは、コンソール・プロセッサおよびランプユニット・プロセッサの間の通信を活動化する。コマンド応答ルーチンは、ランプユニット処理のこのレベルに関係付けをした1つ以上の他のプログラムを活動化する。これらの他の関係付けをしたプログラムは、ステート・データ管理プログラムと、キュー・データ管理プログラムと、通信管理プログラムと、機能論理制御装置と、物理制御管理プログラムがある。これらのプログラムは、多くの場合、ステート・データ管理プログラムにデータを直接報告する。物理制御管理プログラムは、ランプユニットの物理アクチュエータ、例えば、モータと、調光器、等々のアクチュエータを制御する追加プログラムの活動化を監視する。

【0129】コマンド応答ルーチンは、コンソール・プロセッサから発行されたコマンドの実行を必要とするアクションの個々のスクリプトである。このプログラムの流れを図15に示す。ルーチンの中には内部データを操作するものがあり、他のルーチンは指定されたデータをコンソールに伝送し、さらに別のプログラムはランプユニットの物理アクチュエータを動かす必要のある特定のアクションを行い、動かす必要のない場合は、ランプユニットの物理アクチュエータを制御する。上述したルーチンの中には、上述したように指定されたアクションを組み合わせることを要求する。次のコマンド応答ルーチンを記述する際には、コンソール・プロセッサからランプユニット・プロセッサに伝送されるメッセージの第1のワード内に見付け出される値に基づいて、応答ルーチンが選定されることに注意することが重要である。各コマンド・メッセージはコマンド識別子として知られる一意の値を含む。

【0130】第1の関係付けをしたプログラム、すなわち、ステート・データ管理ルーチンは、コンソール・プロセッサおよびランプ・プロセッサの両方からのステータス・データの共通のソースでありレポジトリである。コンソール・プロセッサから受信され、かつ、たまにし

か用いられないデータは、コマンド応答ルーチンにアクセス可能であり、要求時に検索される。より頻繁に用いられるデータは、コンソール・プロセッサから受信された後、機能論理制御装置に渡される。ステート・データというあるデータは、コンソール・プロセッサから、データをバックして、本システム内の各ランプユニットに伝送される。ステート・データは、全てのランプユニットに単一の同時伝送で伝送される。ステート・データ管理プログラムは特定のランプユニットに適用可能なステート・データをこの伝送から抽出する。初期化スクリプト中に、コンソールにより行われる制御チャネル割り当てにより、各ランプに適用可能なデータが識別される。論理および物理制御装置はランプユニットの種々のステートを、直接、ステート・データ管理プログラムに報告する。ステート・データ管理プログラムはランプユニット内の複数ソースからのデータを組み合わせてステータス・データの単一ブロックにする。周期的なコンソール・コマンドにตอบสนองして、各ランプユニットはこのステート・ブロックをコンソールに伝送する。

【0131】通信管理プログラムは、コマンド・インタプリタのオペレーションに関して、上述した関係付けをしたプログラムである。コンソールからの大量データ（RAMベースのプログラムまたはキュー・データ）をダウンロードするコマンド応答ルーチンは、リード(read)コマンドを通信管理ルーチンに発行する。これらのリード・コマンドにより、コンソールから、ランプユニット・メモリの適正なメモリに送信されるデータがストアされるランプユニットがコンソールへのデータ伝送を要求すると、コマンド応答ルーチンは書き込みコマンドを通信管理プログラムに送信する。書き込みコマンドは、ランプユニット・メモリのデータをアクセスするために適正な位置に供給される。

【0132】通信管理ルーチンも、初期伝送がコンソール・プロセッサにより受信されない場合のデータの再送を担当する。データを再送する際に、通信管理ルーチンは大ブロックをフラグメント化し、通信ネットワーク・チャネルのノイズの影響を受けないようにする。

【0133】キュー・データ管理プログラムに関係付けをしたプログラムは、RAMメモリ内に通常のキーインデックス付きファイルシステムを備えている。一意のオペレータ割り当てキュー番号は、キュー・データ・ファイルの各レコードの第1の4バイト中に保持され、そのレコードを識別するためのインデックスとして用いられる。キュー・コール時に、コールされているキューの番号に一致するキュー番号に対し、種々のインデックスが探索される。その結果、探索されたキュー番号とストアされたキュー番号とが一致した場合は、キュー・データ・レコードが検索され、かつ、コマンド応答ルーチンに戻される。一致しない場合は、その旨が同様にコマンド応答ルーチンに報告される。

【0134】ランプユニットはオペレーション可能な機能が多いので、論理制御プログラムがランプユニットの各物理機能に供給される。相互排除するわけではないが、種々のランプユニット機能は、光度と、ポジションと、色と、ビームの論理制御装置を含む。各ランプユニットの物理ハードウェアがこれらの機能を有する方式により、対応する種々の論理制御プログラムがインプリメントされることになる。論理制御プログラムは、物理装置の各機能に対し、単一の制御点を設けることにより、ランプユニットの類似の機能を果たす。全ての論理制御プログラムにより供給されるサービスは、ここではサブマスタという種々のフロントパネル・ソースでリコールされるキュー・データの受信を含む。また、このサービスは他のサブマスタから前回にリコールされたデータと、新たなキュー・データとを統合することと、コンソールから受信されたマニュアル制御コマンドに従って現在の機能データを変更することと、現在の機能データを報告することを含む。論理制御プログラムの中には、現在の機能データをプリセットされた機能値としてストアし、コンソールのコマンドによりこれらのプリセット値をリコールして報告する際に動作するものもある。

【0135】論理制御プログラムの中には、コンソール・プロセッサから送信されたフェーダ値を用い、リコールされたキュー・データをフェーダ値に比例させて変倍させるものもある。物理制御管理関連プログラムは、論理制御装置により計算された現在の機能データを変更させるサブプログラムの活動化を監視する。上述したサブプログラムは2つの主なカテゴリに入る。例えば、ステップモータを制御するサブプログラムは、ステップ・コマンドのタイミング順序をモータに出力する慣用のアルゴリズムをインプリメントする。ステップモータがランプユニット・プロセッサにより伝送されるステップ駆動コマンドに従うことを確認するのに用いるため、ステップモータ・サブシステムは、スイッチ閉路インデックス付きフィードバックを含むことになる。ランプユニットの他の機能は、ランプをバンおよびチルトさせる直流サーボモータを駆動することを含む。直流サーボモータ回転計により出力されるランプ速度情報と、光学エンコーダ/カウンタ回路からのポジション・フィードバック情報とは、システム・データバス上に、ランプユニット・プロセッサへのフィードバック情報として伝送される。これらの構成要素を制御するサブプログラムは、慣用の速度フィードバック・サーボ制御アルゴリズムを利用する。また、ポジション・フィードバック信号が変化すると発生されるハードウェア割り込みによって、サーボ制御ランプ機能が予期せず動いたとき活動化される。サーボ制御ランプ機能の予期しない動きによりランプに生ずる動きであって、ランプユニット・プロセッサにより指令されなかった動きがランプユニット・プロセッサに通知される。当然、これらのサブプログラムに替えてア

ナログまたはデジタル回路を用いることができる。

【0136】そのランプユニットに関するあるステータス・データは、物理制御装置で発信されることになる。例えば、ランプユニット光源の電球の完全性(integrity)は、この光源に電力を供給する電源の振る舞いから取り出される。モータの運動に対応してランプが動かないときは、ランプユニットの運動範囲でランプユニットの動きが妨害されたものと推定する。また、予期したインデックス入力が探索できない場合、ステップモータ・サブシステムに障害が生じたものと推定することができる。このステータス情報は、直接、ステータス・データ・マネージャ・プログラムに報告される。

【0137】上述した2つの例は、マニュアル制御するため、ランプ選択に応答して、コンソール・プロセッサのオペレーションとキュー・データをストアするためのコンソール・プロセッサのオペレーションの例であるが、これら2つの例を次に繰り返し、ランプユニット・プロセッサによるアクションを示す。これら2つの例は、ランプユニット内に生ずる処理を示し、各ランプユニット内の種々のプログラムの活動化と対話を含み、かつ、本発明に係る制御コンソールおよび各ランプユニット間のタスクの配分とを含む。

【0138】第1のランプユニットの例は、コンソール・オペレータがマニュアル制御のためにシステム内の単一のランプを選択したときに生ずる一連のアクションと、ランプユニットの空間的な向きを変えるためのコンソール・デバイスの操作とに関する。両方の例は、必要なRAMベースのプログラムが全てキュー・データとともに、コンソールと完全に同期するものと仮定する。

【0139】メイン・シーケンサ・ループの一部として、ランプユニット・プロセッサはコマンド・インタプリタ・プログラムにジャンプし、通信管理プログラム内の保留のリードコマンドのステータスを検査する。このコマンド・インタプリタ・プログラムは、メモリ空間のブロックを用い通信管理プログラムをサービスする。メモリのこのブロックは、リード・コマンドの実行のステータスを通知するデータの1バイトを含む。リード・コマンドをサービスする際に、コマンド・インタプリタ・プログラムは、通信管理プログラムにより実行されているコマンド・ブロック内のデータのステータス・バイトを検査する。保留のリード・コマンドを完了したこと、すなわち、データのブロックがコンソール・プロセッサから受信されたことを、フラグが示すと、コマンド・インタプリタ・プログラムはこのデータの最初のバイトを調べる。データの最初のバイトの値は、ランプユニットにより実行される特定のコマンドを表わす。

【0140】この例によれば、コンソールから受信されたコマンドは、マニュアル制御channel-selector-mapコマンドであることが分る。このコマンドはコンソールからの追加のデータを必要としないので、コマンド・イン

タブリタ・プログラムは別のリード・コマンド・ブロックを確立し、通信管理プログラムを再び活動化する。ついで、通信管理プログラムはランプユニットに別のコンソール・コマンド伝送を受信する準備をさせ、コマンド・インタブリタに戻る。ついで、コマンド・インタブリタ・プログラムは関係付をしたコマンド応答ルーチンにジャンプする。上述したコンソール・コマンドは、ネットワークに伝送され、かつ、このネットワークとコネクションした全てのランプユニットにより同時に受信されるメッセージを表わす。特に、特定のランプユニットに関して上述した処理は、本システムの他のランプユニットにも並行して生じていることになる。

【0141】本発明の舞台照明システムは、1000個以上の舞台照明灯に適用することができるから、データの複数バイトをネットワークを介して伝送しなければならない。第1ビットは、特定のランプユニットに対応しており、このビット位置は、ランプ・ステータス初期化スクリプト中に、コンソールによりランプユニットに割り当てられたコンソール制御チャンネル番号から導かれる。本システムの他のランプユニットは、異なるコンソール制御チャンネル番号が割り当てられる。各ランプユニットはそれ自体のビットデータを125byteのブロックから独立して抽出することになる。コンソール制御チャンネル番号は、ステート・データ管理プログラムにストアされる。

【0142】コマンド応答ルーチンの必要とするアクションは、コンソール伝送を復号化した結果、125byteのブロックのメモリに記憶ロケーションを有するステート・データ管理プログラムにジャンプすることである。コマンド応答ルーチンは、マニュアル制御ステータス・ビットが操作されるビットであることを示す識別子を供給する。

【0143】ステート・データ管理プログラムは、コンソール制御チャンネル番号をインデックスとして利用し、マニュアル制御のためのランプユニット選択/選択解除に関するブールフラグに割り当てられた値を抽出するサブプログラムを有する。このブールフラグは、マニュアル制御コマンドが受信されたときに参照され、ランプユニットによる応動を許可するか、あるいは許可しないのいずれかである。ついで、ステート・データ管理プログラムからの制御は、メイン・シーケンサの永久ループに戻される。

【0144】メインシーケンサ・プログラムの命令を実行するランプユニット・プロセッサは、コマンド・インタブリタ・プログラムに周期的にエンタし、新たな伝送がコンソールから受信されたか否かを確認する。コンソール・オペレータがランプ・ポジション調整つまみを操作していることを示すコマンドがコンソールから受信されたものと仮定する。よって、コマンド・インタブリタにより受信された次のランプユニット・コマンドはエン

コード変更コマンドの最初のバイトから判定される。ついで、適正なコマンド応答ルーチンにジャンプする。再び、このコマンドがネットワーク内の全てのランプユニットにより同時に受信され、しかも、このようなランプは全て適正なアクションを並行して実行することになる。

【0145】上述したコマンドに関するコマンド応答ルーチンは、まず、ステート・データ管理プログラムにより、ブールフラグが、現在、マニュアル制御のための特定のランプユニットを選択するか、あるいは選択解除するかを示すか否かを検査する。フラグがセットされない場合は、コマンド応答ルーチンは終了し、ついで、ランプがマニュアル制御のために選定されないと、エンコード変更コマンドが無視される。しかし、この例では、フラグが先行するマニュアル制御チャンネルセレクトマップ・コマンドの一部としてセットされるものと仮定されると、処理が継続される。

【0146】コンソール制御のポジションの変更に応答して、コンソールにより伝送されるエンコード変更コマンドは、特定のコンソール・エンコードを識別する1byteのデータを伴う。このことは、コンソールパネルに幾つかのエンコードがあるので、極めて重要なことである。各エンコードは異なるランプ機能の制御を行う。さらに、エンコード変更コマンド・バイトはエンコード入力値の変化量を表わすデータを含む。各エンコードにはランプユニットの異なる機能が関係付けをしてあるので、コマンド応答ルーチンは、変更された入力値を有するエンコードと関係付けをした機能論理制御装置へのジャンプを実行する。コマンド応答ルーチンはランプユニットのポジションを変えるだけの量に対応するデータも渡す。

【0147】この例で、ポジション論理制御装置が活動化される。このポジション論理制御装置は、ランプユニットの現在のコマンドを表わすデータを、ランプユニットのポジションと解釈し、このデータを、受信されたエンコード変更入力値に正比例する量だけ修正する。この新たな値はランプユニットの新たなポジションとしてストアされ、ついで、ポジション論理制御装置はコマンド・インタブリタに戻る。

【0148】次に、コマンド・インタブリタ・プログラムは物理制御管理プログラムを活動化する。物理制御管理プログラムは、全ての論理制御装置により、メモリにやがてストアされるコマンド・データを、ランプユニット物理デバイスの実際のポジション・ステートと比較する。物理デバイスの実際のステートを、指令されたステートと一致させる。2つ以上の機能データが変更される状態では、物理制御マネージャは、全ての物理アクチュエータが適正にパフォーマンスすることを保証するため、物理アクチュエータのプログラムを、プリプログラムされた組み合わせで活動化することになる。

【0149】この例では、サーボモータ制御プログラムのみが活動化される。このプログラムは、新たなコマンド・データにより要求される変化方向を計算し、同様に、サーボモータに印加される電圧の適正值を計算する。また、関係付けをしたタイマがトリガされ、周期的にハードウェア割り込みを行う。割り込みごとに、サーボモータ制御プログラムは、サーボモータ・サブシステムの実際のステートがポジション論理制御装置により確立されたコマンド・データと一致するまで、モータに印加する適正な電圧を再計算する。

【0150】このようにして、所望のサーボモータを回転させて、例えば、ランプユニットのパンまたはチルトのポジションを対応させて変化させる。サーボモータの回転が開始されると、ランプユニットは、サーボモータ制御プログラムおよび物理制御管理プログラムから、コマンド応答ルーチンおよびコマンド・インタプリタに戻る。制御はコマンド応答ルーチンおよびコマンド・インタプリタからメイン・シーケンサに戻る。メイン・シーケンサでは、受信コマンド、メモリ・チェックサム障害および通信アドレス変更処理の走査が再開される。新たなランプユニット・ポジションに到達するようなときまで、ハードウェア割り込みおよびサーボモータ制御再計算は、メイン・シーケンサの永久ループのアクションにより散在される。コンソールによりネットワークを介して各ランプユニットに伝送される高レベル・コマンドは、各装置で、各装置に対するコマンドの影響を判断し、利用可能な場合には、所望の結果を得る追加処理を行う。

【0151】次の例には、ランプユニットでの処理が、コンソールパネルの「ストア・キュー」スイッチをコンソール・オペレータが操作した結果として含まれる。ランプユニット・プロセッサは、メイン・シーケンサの永久ループを終了し、コマンド・インタプリタにジャンプし、通信管理プログラム内の保留リード・コマンドのステータスを検査する。この例では、コマンド・インタプリタ・プログラムは、コンソールから新たに受信されたメッセージを見付け出す。コマンド・メッセージの最初のバイトには、ストア・キュー・オペコードを有する。コマンド・インタプリタは、通信管理プログラムに対してリード・コマンドを再スタートさせ、ストア・キュー・コマンド応答ルーチンをコールする。このコマンドはネットワークの全てのランプユニットで同時に受信され、このようなユニットは、全て、アクションの次のシーケンスを並行に実行する。

【0152】コマンド応答ルーチンでは、各論理制御装置が現在指令されている機能データについて照会する。このデータは10 byteのメモリ記憶区域にパックされる。さらに、このデータのブロックは、キューに対してオペレータにより割り当てられた番号を表わす4 byteのデータと組み合わせられる。当然、このキュー番号はコン

ソールからのキューストア・コマンド伝送の一部として受信される。ついで、コマンド応答ルーチンはキューデータ管理プログラムを呼び出し、これとともに、上述した処理から得られるデータの14 byteのブロックを転送する。

【0153】キューデータ管理プログラムは、コマンド応答ルーチンにより渡されるレコードキュー番号のインデックスと一致するインデックスに対し、そのレコード・インデックス、すなわち、キュー番号のリストを走査する。走査の結果、一致している場合は、データレコードがコマンド応答ルーチンから受信されたデータレコードによりオーバーライトされる。探索中にインデックスが一致しない場合は、インデックスおよびデータのファイル内の空白のレコード区域に、新たなレコードが書き込まれる。ランプ装置のこのデータメモリは、コンソール・ディスク・コピーにまだ伝送されておらず、現在のランプユニット・キューデータを更新する種類のものである。データ・レコードをコンソール・ディスク記憶装置に伝送するのが遅れた場合には、複数のキュー・レコードに対し十分な余地を設けなければならない。ついで、キューデータ管理プログラムはコマンド応答ルーチンに戻る。

【0154】コマンド応答ルーチンはステート・データ管理プログラムを即時に呼び出し、フラグをランプ・ステータス・ワードにセットする。このランプ・ステータス・ワードはランプユニットがコンソール・ディスク記憶装置への伝送に対してレディ状態のキュー・データを有することを示す。ついで、ランプユニットがメイン・シーケンサに戻るまで、プログラムは互いに他のプログラムにより順次に戻される。ついで、メイン・シーケンサの永久ループ内の処理が再開される。

【0155】キュー・ストア・コマンドの処理では、幾つかの点で、コマンド・インタプリタは、ランプステータス・レポート・オペコードを有するメッセージの受信をセンスする。予備のリード・コマンドが通信管理プログラムに再発行され、そして、lamp-status-reportコマンド応答ルーチンがコールされる。これらのlamp-status-reportコマンドがネットワーク内の各ランプユニットに個々にディスパッチされる。この場合、1度に1個のランプユニットのみがコンソールに応答する。

【0156】コマンド応答ルーチンはステート・データ管理プログラムをコールし、ランプ・ステータス・ワードの現在値を得る。メモリのこのブロックは、通信管理プログラムに発行された書き込みコマンド内のメッセージ・データとして利用される。この書き込みコマンドは、次のコンソール・コマンド伝送の準備に際して丁度パフォームされたリード・コマンドとは対話しない。種々のプログラムによりメイン・シーケンサに戻り、そこで永久ループが再開される。

【0157】ランプ・ステータス・データにセットさ

れ、かつ、前回のコマンドでコンソールにレポートされたフラグは、コンソールをプロンプトし、read-cue-data-change-buffer コマンドを発行する。このコマンドは、ランプユニットの通信管理プログラムにより受信され、コマンド・インタプリタ・プログラムによりセンサされる。さらに、このコマンドは特定のランプにアドレッシングされ、このランプのみが応答を伝送することになる。read-cue-data-change-buffer コマンド応答ルーチンは、キュー・データ管理プログラムから、新たなキュー・データのリストを検索する。さらに、上述したコマンド応答ルーチンは、ライト・コマンドで、リストをメッセージデータとして通信管理プログラムに送信し、ステート・データ管理プログラムをコールし、キュー・データ変更バッファ内にデータの存在することを示すフラグをクリアする。その結果、ランプユニット・プロセッサは、メイン・シーケンサの永久ループに戻り、別のコンソール・コマンドを待つ。

【0158】以上、舞台照明灯のポジションを変更するのに必要なランプユニット・プロセッサのアクションを説明し、同様に、ランプユニット・メモリ内のキュー・データのストアを説明した。しかし、本システムの適応は上述したことに限定されない。各コマンドは完全に説明する必要がない、本発明の妨げになるだけである。本発明の各ランプユニットに関連して用いられる他のランプ・コマンドを次に挙げる。

【0159】本発明に係るオペレーションを行うプログラムの全機能は、図10ないし図15に詳細に記載してある。全プログラムの代表部分に対してリストした詳細なコードを次に示す。これは図15を参照して記載された色論理制御をインプリメントするのに必要なコードである。このコードはMotolora製のマイクロプロセッサ68000型で実行するために書かれている。色論理制御プログラムは、光度と、ポジションと、ビーム径に対する論理制御プログラムと全く同様である。

【0160】開示した照明システムにより、数百個の自動化ランプユニットが正確で効率的でかつ柔軟に制御されることは、上述したことから明らかである。ランプユニットからのステータス・データが制御コンソールにレポートされる。このステータス・データには、パラメータ・データであって、ランプユニットの現在の光度と、色と、ビーム形状と、ビーム方向を含み、同様に、リコールされている現在のキューと関係付けをした任意のタイミング・パラメータを含むパラメータ・データを、リアルタイムに表示することを含めることができる。ランプユニット内のオペレーティングシステム・プログラムと関係付けをしたメモリで、重大な論理誤りを経験したオペレーティングシステム・プログラムは、更新される。また、種々のキューに関係付けをしたパラメータ・データがストアされる。このパラメータ・データをストアすることにより、オペレータはショー(show)を行うのに用いられ

るデータをセーブすることができ、しかも、このデータを、同様に構成されているが別々の異なるランプユニットによりなる照明システムに、ロードすることができる。これらランプユニットを、前回行われたショーの物理的なロケーション、例えば、異なる大陸での物理的なロケーションとは異なる物理的なロケーションに配置させることができる。

【0161】図16を説明する。簡単なデータ・リピータ回路(詳細は図9に示す)は、ブロードキャスト・ネットワーク38に接続されたアクティビティ・センサ392と、応答ネットワーク40に接続されたアクティビティ・センサ394を含む。各アクティビティ・センサ392、394は赤LED396を(LEDが明滅して見えるようにするパルス伸長回路を介して)駆動する。これらのLEDは、リピータ・ボックスの外側に取り付けられ、リピータが接続されている対応するネットワークのブランチ(branch)で、電気的なアクティビティがあるごとに明滅する。

【0162】また、簡単なデータ・リピータはManchesterデコーダ/エンコーダ352を含む。Manchesterデコーダ/エンコーダ352はブロードキャスト・ネットワーク38およびManchesterデコーダ/エンコーダ374に接続されている。Manchesterデコーダ/エンコーダ374は応答ネットワーク40に接続されている。既に説明したように、Manchesterデコーダ/エンコーダ集積回路は「リピータ・モード」でコネクションさせることができる。この「リピータ・モード」では、Manchesterデコーダ/エンコーダ集積回路の入力端子で受信されたメッセージはデコードされ、そして、再コード化され、さらに伝送される。各エンコーダ/デコーダは緑LED398を(パルス伸長回路を介して)駆動する。緑LEDもまたリピータ・ボックスの外側に取り付けられ、Manchester符号化されたデータがリピータに入力されるごとに点灯する。正常状態では、赤および緑LEDは同時に明滅する。他の任意の条件、例えば、赤LEDが明滅するが、緑LEDが明滅しない場合は、データ伝送に誤りがあることを示す。しかし、このように構成しても、ネットワークにより伝送されるメッセージの複雑な誤りは検知することができない。

【0163】図17および図18に示すように、この自動化ステージ照明システムにインプリメントされたデータ通信ネットワーク26は、幾つかの介入させたリピータ回路を介して、複数の自動化ランプユニット(automated lamp unit; ALU)と通信を行う制御コンソール24を含む。電力および信号分配ラック400に配置された「コンソール・リピータ」54は、制御コンソールから受信し、1つ以上の「トランク・リピータ」55へのこれらの通信を繰り返す。1つのこのようなトランク・リピータ55は、コンソール・リピータ54を有する分配ラック400に通常配置されている。

【0164】トランク・リピータは、7つのトランクケーブル・コネクタ402に対するデータ通信リンク26Cをサポートする。データ通信リンク26Cは、適正なマルチ導体トランクケーブルにコネクションされると、電力とデータを、ランプユニットの近傍の通常の照明トラスに掛けてあるリピータ・ボックスに供給する。また、コンソール・リピータ54は8つの補助データ・コネクタ404に対するリンク26Bをサポートする。補助データ・コネクタ404は、リンク26Bを用いて、適正なデータ入力コネクタ406を介して、追加分配ラック408に接続される。よって、データ信号は、追加分配ラックに配置されているトランク・リピータ55に供給される。ついで、追加分配ラックは電力とデータを、ランプユニットの近傍の通常の照明トラスに掛けてある他のリピータ・ボックスに供給する。そして、各リピータ・ボックスは電力とデータを最高9つのランプユニットに供給する。

【0165】照明制御システムの一実施例では、1つの制御コンソール24が1つの「マスタ」分配ラック400に接続され、8つの「スレーブ」分配ラック408に、補助データ出力コネクタ404を介して接続されている。各分配ラックはトランクケーブルを介して7つのリピータ・ボックスに接続されている。そして、各分配ラックは電力とデータを最高63個のランプユニットに供給することができる。そして、1つのマスタラックと8つのスレーブラックは、電力とデータを最高567個のランプユニットに供給することができる。このシステムの能力を、システムソフトウェアによりサポートされる1000個のランプユニット構成まで拡大するため、各スレーブラック408は、トランク・リピータの予備出力により駆動されるデータ出力コネクタ410を介して、追加スレーブラックに接続することができる。このようにリンク26Eを介して接続された8つの追加スレーブラックは、電力とデータを最高504個の追加ランプユニットに供給する。すなわち、システムソフトウェアによりサポートされる1000個のランプユニットを優に超えるランプユニットに供給する。

【0166】図19に示すように、ブロードキャスト・ネットワークは同一のデータ信号を実際的には同時に全てのランプユニットに供給する。ブロードキャスト・ネットワーク38を介して、コンソール24は各メッセージを各ランプユニットの各受信機(RX)に送信する。

【0167】図20は応答ネットワーク40の相互結合を示す。コンソール24はメッセージをブロードキャスト・ネットワーク38を介して第1ランプユニットに送信することにより、ランプユニットからステータス・データを要求し、そして、応答ネットワーク40を介するそのランプユニットの応答を待つ。そのランプユニットからのステータス通知メッセージがコンソールにより受信された後、そのシステムの他のランプユニットに対し

てそのシーケンスを継続することができる。ランプユニットが、メッセージを送信するトランスミッタ(TX)と、メッセージを受信するレシーバ(RX)とを含むことを除けば、応答ネットワークはブロードキャスト・ネットワークと同様にして接続されている。

【0168】各応答伝送の間、ランプユニットおよびリピータ・ボックスの間の多くのリンク40Dのうちの1つのみが利用される。図20に示すように、リピータ・ボックスおよび分配ラックの間の1つのリンク40Cのみと、トランク・リピータおよびコンソール・リピータの間の1つのリンク40Bのみと、コンソール・リピータおよび制御コンソール24の間の1つのリンク40Aのみとを介して、応答伝送がそのコンソールに到達する。従って、1つのランプユニットからステータス・データを獲得するのに1単位時間が必要な場合は、1000個のランプユニット全てからステータス・データを獲得するには1000単位時間を要する。

【0169】2つ以上のランプユニットがステータス・データに対する1つの要求に応答する場合、コンソール・リピータおよび制御コンソールの間のリンク40A上に、マルチプル伝送が同時に現れることは、当然のことである。同様に、応答ネットワークに注入された任意のノイズは、コンソール・リピータおよび制御コンソールの間のリンク40A上の許可された信号の上にスーパインポーズされ、その結果、コンソールにより誤って受信される。本発明の1つ以上の態様に係る改良されたリピータによれば、誤ったランプユニット伝送および応答ネットワークのノイズが乗ったリンクを識別するとともに、分離することができる。

【0170】図21に示す改良されたリピータは、プロセッサ450と、このプロセッサに関係付けをしたROM(read-only memory)と、RAM(random-access memory)とを含むとともに、アクティビティ・センサおよびManchesterデコーダ/エンコーダからの入力を受信する制御回路を含む。プロセッサ450はこれらの入力を変換し、プロセッサ450と関係付けをした制御回路により、LED396、398を点灯させ、データリンク・ネットワークの条件を示す。例えば、緑LEDの点灯はデータリンク・ネットワークが適正に動作していることを示し、一方、赤LEDの点灯はデータリンク・ネットワークが誤動作していることを示す。アクティビティが少しもない場合は、赤LEDおよび緑LEDを消灯させることができる。赤LEDおよび緑LEDの対は、個々に、ブロードキャスト・ネットワークおよび応答データリンク・ネットワークに供給することができる。あるいはまた、英数字・ディスプレイ・デバイス452をリピータ回路に組み込み、簡単なコードおよびメッセージを表示する。

【0171】図22に示す別の改良されたリピータは、コンソールおよびランプユニット通信回路で用いられて

いるようなマルチプロトコル通信制御装置チップ454に組み込まれている。Manchesterデコーダ/エンコーダに結合された通信制御装置チップを用いて、有効な通信制御装置割り込みをしないライン・アクティビティを直ちに検知することができる。図示したその追加ゲート456およびマルチプレクサ458により、プロセッサは個々の応答ライン入力をサンプリングすることが可能であるが、ある応答ライン入力により、ノイズを停止させるか、あるいは不許可の伝送がシステムの他の部分に広がるのを停止させることは不可能である。

【0172】オペレーションの1つのモードでは、ゲート456に出力される信号が9入力論理ORゲートに印加され、組み合わせられて1つの信号にされ、ライン462上に出力される。ゲート456への9つの入力のうちの1つのみが、適正に稼働しているシステムでは、任意の時点で同時に活動化されることになるので、1つの信号のみがライン462上に同時に現れることになる。ライン462上の信号が論理ゲート466を介してManchesterデコーダ468に結合され、その後、通信制御装置454に印加される。通信制御装置454では、信号の誤りをプロセッサ450により検査することができる。誤りが検査されない場合は、プロセッサおよび通信制御装置は、メッセージをManchesterエンコーダ470を介して応答ネットワーク40の次のブランチ上に伝送する。応答ネットワーク40上で受信された信号に誤りが検知されない場合は、診断モードがリピータ・プロセッサ450により入力される。制御バス476として図示される複数の論理制御信号を用いて、プロセッサ450はマルチプレクサ458を動作させ、ORゲート460の入力で種々の離散信号をサンプリングする。ライン464上のマルチプレクサ458の出力はゲート466に印加される。ゲート466は制御バス476を介して動作され、マルチプレクサ458をManchesterデコーダ468に接続する。

【0173】誤り検知モードで、マルチプレクサ458のオペレーションと通信制御装置454とを協調させることにより、プロセッサに結合されたランプユニットのうちの1つのランプユニットが、不明瞭な信号またはノイズを伝送しているか、あるいは不適正な時点で伝送して他の適正な信号を誤って伝送することになることを、プロセッサ450は判定することができる。そして、プロセッサは制御バス476を利用し、ゲート456の誤りのある入力をディスエーブルする。よって、適正にオペレートしているランプユニットに対する通信の完全性を回復させる。

【0174】ブロードキャスト・メッセージは同様に処理される。ブロードキャスト・ネットワーク38上に出力される信号は、Manchesterデコーダ472に印加され、その後、通信制御装置454に印加される。通信制御装置454では、信号の誤りをプロセッサ450に

より検査することができる。

【0175】本発明の別の機能によれば、図23に示す「スマート・リピータ」は、通信制御装置454およびRAM482の間に接続されたDMA (direct memory access) 制御装置480を含む。その構成は、機能的には、制御コンソールおよびランプユニットに用いられるプロセッサ/モデム複合体と同一である。このような回路構成による利点の1つには、各スマート・リピータが全く任意のランプユニットのように、そのコンソールと通信を行うことができることにある。

【0176】全てのリピータにより実質的に同時に受信されたネットワーク制御メッセージを、コンソールは送信することができる。ネットワーク制御メッセージを特定のリピータ装置にアドレッシングすることができるか、あるいは、ネットワーク制御メッセージを共通のリピータ・アドレスを用いて全てのリピータ装置にアドレッシングすることかできる。各リピータ装置は、そのアドレスまたはメッセージの内容により、個々に、そのメッセージに回答する。例えば、リピータを命令してランプユニットのステータス・ポーリングを開始させるメッセージは、共通のリピータ・アドレスに送信されることになる。特定のリピータを命令してランプユニット・ステータス・データのブロックをコンソール（または、応答ネットワーク上の次のリピータ装置）に伝送するメッセージは、特定のリピータ・アドレスに送信されることになる。リピータもネットワーク・ステータス・メッセージを必要ととき送信する。そのメッセージは、例えば、検知された誤りの種別を表し、ネットワークのブランチが誤りを表し、ブランチはディスエーブルされているデータを含む。

【0177】オペレーションの1つのモードでは、ブロードキャスト・ネットワーク38に出力される信号はアクティビティ・センサ392により検知され、Manchesterデコーダ472によりデコードされる。そして、それらの信号はバイパス・ゲート484を介して通信制御装置454にルーチングされる。DMA制御装置480および通信制御装置454は信号を受信しRAM482に入力する。DMA制御装置480および通信制御装置454では、デコードされたメッセージを、検査するか、あるいはプロセッサ450により変換することができる。誤りが検知されず、かつ、メッセージがランプユニットに関する情報を含む場合、プロセッサは新しいメッセージを構成するか、あるいは元のメッセージをランプユニットに再送することができる。そして、DMA制御装置480および通信制御装置454は協調して、メッセージをバイパス・ゲート484を介するとともにManchesterエンコーダ474を介して伝送する。Manchesterエンコーダはゲート478によりブロードキャスト・ネットワーク38に接続されている。直ちに再構成され、ゲート456および478をオペレートさせる制御バス

476を用いて、プロセッサ450はブロードキャスト信号をゲート478の9つ全ての出力端子から出力させるか、あるいは、9つの出力のうちの任意の1つ以上の個々の出力から出力させる。また、制御バス476は入力ゲートを操作し、選択された個々の入力を上述した方法によりイネーブルまたはディスエーブルさせることができる。

【0178】応答モードで、応答ネットワーク40から受信された信号に誤りが検知されない場合は、リピータ装置はランプユニットに要求を出してメッセージを再び伝送させることができる。数回要求を試みた後、リピータが特定のランプユニットから誤りのないメッセージを受信することができないか、あるいは、リピータ・プロセッサがプロセッサに接続された2つ以上のチャネル上に誤りを検知した場合、診断モードがリピータ・プロセッサにより入力される。誤りが検知されない場合、プロセッサおよび通信制御装置はManchesterエンコーダ474を介して応答ネットワーク40の次のブランチ上にメッセージを伝送する。

【0179】スマート・リピータから得られる他の改良点としては次の点を含む。すなわち、有効な通信制御装置割り込みとならないライン・アクティビティを検知する点と、フレーミング誤りか、巡回冗長検査(CRC; cyclic redundancy check)誤りか、あるいは、通信制御装置により検知されるオーバーラン誤りを受信する点と、通信ソフトウェアによる各メッセージに付加されるヘッダデータの誤りを検知する点と、幾つかのデータメッセージの論理誤りを検知する点と、伝送されたメッセージに応答して否定応答(NACK)または肯定応答(ACK)を受信する点と、応答ライン入力をディスエーブルしてノイズまたは不許可の伝送がネットワークの他の部分に広がらないようにする点と、複数のランプユニットからのステータス・データまたは他のリピータからのステータス・データを収集する点と、オペレーティングシステム・プログラムをランプユニットまたはリピータにダウンロードする点を含む。

【0180】上述した改善点から得られる利点としては、次の1)~4)を含む。

【0181】1) システム技術者への改善されたフィードバックであって、データリンク表示器をより視認し易くすること。

【0182】2) センスされた誤りのロケーションをコンソールにレポートして中央操作室に表示すること。

【0183】3) (通信誤りがある) 低下モードでのシステムの改善された操作能力。

【0184】4) 正規モードでの通信サブシステムの改善されたスループット、特に、ステータス・ポーリングに関係するスループット。

【0185】本発明に係るスマート・リピータは、図23に示すように、通信リンクの利用と、全てのランプユ

ニットからのデータを収集するのに必要な時間の両方を劇的に改善することができる。一度、プロセスを開始するメッセージ、例えば、共通のリピータ・アドレスへのメッセージ・ブロードキャストを、制御コンソールが送信すると、トラス・リピータ56は、全て、各トラス・リピータに接続されている9つのランプユニットからデータを同時に収集する。そして、トランク・リピータは、全て、各トランク・リピータ55に接続されている7つのトラス・リピータ56からデータのブロックを同時に収集する。そして、マスタ分配ラック400のコンソール・リピータ54は、コンソール・リピータ54に接続されている9つのトランク・リピータ55からデータのブロックを収集し、収集された全てのデータのブロック全体を、コンソール24に1つのメッセージとして送信する。

【0186】1つの実施例によれば、通信リンクが益々利用される。というのは、63個のトラス・リピータ56は任意の時点で同時に63個のリンク26Dを用いているからである。567個のランプ・ユニットからのステータス・データを収集してトラス・リピータに入れるには、9単位の時間のみを必要とする。その後、9つのトランク・リピータ55は9つのリンク26Cを同時に用いている。63個のトラス・リピータからデータを収集しトランク・リピータに入れるには、7単位の時間を必要とする。1つのコンソール・リピータ54は依然として同時に1つのリピータ26Bのみを用い、9つのトランク・リピータからステータス・データを収集するには、9単位の時間を必要とする。

【0187】コンソール・リピータ54からの1回の伝送で、制御コンソールが、567個のランプ・ユニットからのステータス・データを応答リンク40Aを介して受信することは、より意義のあることである。よって、566個のメッセージ・ヘッダを伝送するのに必要な時間を節約することができる。同一量のデータが非常に小さいオーバーヘッドで伝送される。したがって、改善されたランプ対コンソール応答プロセスにより、ステータス・レポートを収集するのに必要な時間と、誤り確率とが著しく減少する。さらに、ランプ・ユニットはデータをトラス・リピータに伝送するが、トランク・リピータはデータをコンソール・リピータに伝送する。トラス・リピータはデータをトランク・リピータに伝送するが、コンソール・リピータはデータをコンソールに伝送する。よって、データリンクが益々利用される。このように、スマート・リピータはそれ自体のステータス情報をランプ・ユニット・ステータス・データの集合にインタリーブする。

【0188】本発明に係るスマート・リピータは、そのスマート・リピータに接続された全てのランプ・ユニットに対するオペレーティングシステム・プログラムを保持し、システム全体と関連させないで必要に応じてダウ

ンロードする。オペレーティングシステム・プログラムのストアおよびダウンロードは、各ランプ・ユニットの構成により行われる。トラス・リピータによりこのようなダウンロードが行なわれる場合は、トラス・リピータに接続された他の8つのランプ・ユニットのみが、ダウンロードの間、任意のシステム・キュー・コマンドを受信することができないが、システムの他の部分は自由に正常にオペレートすることができる。さらに、全てのランプ・ユニットがオペレーティングシステム・プログラムをダウンロードする必要がある場合、照明トラスに掛けた7つのスマート・リピータは、制御コンソールより大幅に少ない時間でオペレートすることができる。

【0189】スマート・リピータは、図23に示すように、種々のブロードキャスト・リンク出力を出力するゲート478のセットと、種々の応答リンク入力を入力する個々のゲート456のセットとを含む。このように配置することにより、スマート・リピータは、個々に選定されたランプ・ユニットと通信することができる。例えば、2つのランプ・ユニットがたまたま同一アドレスに設定されている場合、ステータス要求を受信したとき、両ランプ・ユニットがステータス・レポートを伝送することになる。その結果、リピータ側で誤って受信することになる。ついで、スマート・リピータは各出力を個々に伝送し、スマート・リピータに接続されているランプ・ユニットから、そのランプ・ユニットに割り当てられたIDまたはアドレスを要求し、対応する入力端子で応答を受信する。2つのランプ・ユニットに同一アドレスが設定された場合、スマート・リピータはこのことがたまたま生じたか否かを判定し、その情報をコンソールに報告し、オペレータに表示する。スマート・リピータそれ自体はフォームおよびファンクション・スイッチの設定により識別することができる（プロセッサをランプ・ユニットではなくリピータとして識別する）。また、スマート・リピータ自体はつまみホイールスイッチを設定することにより識別することができる（どのリピータがプロセッサかを識別する）。ファンクション・スイッチおよびつまみホイールスイッチは、リピータ装置一致回路494に含まれている。あるいはまた、コンソール・リピータ54はコンソール・リピータ54に接続されている各トランク・リピータ55にIDを割り当てることができ、そのIDをその9つの出力端子を介して1度に1つだけ伝送する。その後、各トランク・リピータ55はトランク・リピータ55に接続されている各トラス・リピータ56にIDを割り当て、そのIDを各出力端子を介して1度に1つだけ伝送することができる。

【0190】プロセッサ制御デバイスを利用した任意のシステムは、プロセッサ・ロックアップの確率や、壊されたデータによりプロセッサが正常な機能を果たせない条件や、プログラム命令の永久ループを不注意に実行す

ることに適合しなければならない。本発明に係るスマート・リピータはこの確率を予想し、通信制御装置に関係付けをした論理ゲート484のセットに供給する。スマート・リピータはManchesterデコーダおよびエンコーダおよび通信制御装置の間の信号をルーチングする。デフォルト状態では、バイパス・ゲートはブロードキャスト・デコーダ472から出力されブロードキャスト・エンコーダ474に入力される信号をルーチングする。一方、応答デコーダ468から出力され応答エンコーダに入力される信号をルーチングする。各デコーダおよびエンコーダはそれ自体デフォルト状態では「リピータ」として接続され、入力端子に出力された信号を再コード化し、信号をその出力端子に供給する。初期パワーアップ時での信号リピータ装置のデフォルト状態は、「ダム(dumb)リピータ」の状態であり、ダム・リピータのオペレーションは図9に示すリピータ52のそれと同様である。

【0191】スマート・リピータのプロセッサが初期化され、ストアされたプログラムの実行を開始すると、1つの周期的な機能が、バイパスゲート484に組み入れられているハードウェア・タイマを、制御ライン486を介してリセットし、かつ、その1つの周期的な機能がゲートを切り換える。その結果、全ての信号が通信制御装置によりルーチングされる。バイパス・ゲートに関係付けをした制御論理ゲートは、論理信号488を生成し、生成された論理信号488は、Manchesterデバイスに印加され、そのデバイスをリピータから必要なエンコーダまたはデコーダに再構成する。プロセッサが正常に機能し、かつ、バイパスゲート484に関係付けたハードウェア・タイマを周期的にリセットする限り、その装置はスマート・リピータとして機能する。プロセッサがそのプログラムの適正な実行に失敗し、かつ、その実行を中止した場合、タイマはタイムアウトし、その装置はダム・リピータ・モードに切り換えられる。ダム・リピータ・モードのデフォルトまたは緊急オペレーションに対して取られる対策は、信号リピータ装置の1つでのプロセッサ障害が生じた場合でも、システムデータ通信ネットワークの連続性(continuity)を確認する。

【0192】本発明のプロセッサ制御リピータは、どれも、標準シリアルデータ・ポート490を有し、ポータブルまたはハンドヘルドのデータ端末に接続することができる。技術者はこのような端末を、リピータ・ボックスまたは分配ラックに設けたシリアルポート・コネクタに接続することができる。また、技術者はその端末を用いてデータリンク・システムの診断検査を始め、しかも、検査結果および/またはステータス・データを受信することができる。例えば、1つのリピータ・ボックスの赤応答リンクLEDが点灯された場合、技術者はハンドヘルド・データ端末を用いてリピータ・ボックスにブラグインし、表示された誤動作のより詳細な情報を受信

する。また、技術者は端末を用いてさらに検査を始めることができる。この検査を、リピータのプロセッサにより実行することができる。または、そのリピータから送信されたメッセージにより制御コンソールの検査を要求することができる。

【0193】ポータブル・データ端末は、システム・コマンドを処理する時点と、データリンク・ネットワークを介して伝送されるランプ・ユニット応答との間の予備時間で、リピータ・プロセッサと通信することができる。ポータブル・データ端末を用いる技術者はメッセージをコンソールに伝送し、1つ以上のランプ・ユニットに送信すべきシステムコマンド・メッセージを要求する。技術者は照明リグで稼働している間に電球をこのように点灯させるとともに消灯させる。あるいはまた、端末を用いた技術者は、単一のリピータ装置に接続されている1つ以上のランプ・ユニットに、メッセージを伝送することができる。分配ラックに接続されている端末は、その端末に接続されている複数のトラス・リピータのうちの1つ以上のトラス・リピータに、メッセージを伝送することができる。

【0194】ポータブル・データ端末の代替端末として、スマート・リピータは英数字キャラクタ・ディスプレイ452を含むことができる。英数字キャラクタ・ディスプレイ452はエラーコードまたは同様の人間が可読なメッセージを表示することにより、データリンク・ネットワークのステータスを示す。複数の押ボタン・スイッチ492を入力デバイスとして提供することができ、プロセッサによりディスプレイに書き込まれた単一の入力選択メニューに関連させて、複数の押ボタン・スイッチ492を用いることができる。このように、技術者はエラーコードの報告を要求し、単一の組み込まれたデータ端末により、リピータ装置プロセッサと通信を行い、診断ルーチンまたは他の機能を開始することができる。

【0195】図17および図18に示し、既に説明したように、制御コンソール24と種々のランプユニットの間のデータリンク・トラフィックが、分配ラック400の制御コンソール24とコンソール・リピータ54の間のリンク26Aを介して、移動する。好ましい実施の形態では、ハードウェア障害があった場合、図17に示すデュープリケート・リンク580が、あるコネクションにより、バックアップ制御コンソール582に接続される。しかし、このデュープリケート・リンク580は、活動化されるまで休眠状態にあり、メイン・データリンク26Aが既に供給している以上のデータ通信容量は提供しない。

【0196】好ましい実施の形態では、本明細書で開示されている照明システムは、モジュラ制御システムにより制御されている。これは、制御システム全体に影響を与えないで、必要に応じて、ソフトウェアおよび/また

はハードウェアの能力を高めるために、個々のシステム・コンポーネントまたはモジュールのアップグレードまたは置換を容易にするためである。さらに、必要に応じて、モジュラ制御システム500を再構成して、異なるショー(show)の様々な要件に適合させるため、個々のモジュールは交換可能であるのが好ましい。図26および図27に示すように、モジュラ制御システムはモジュラ・コントローラ・メインフレーム500を備えている。このメインフレーム500は、制御パネル・ユニット546-552と相互結合され、制御資源ネットワークの他の制御デバイスと、入力モジュール590のセットにより相互結合されている。入力モジュール590は、出力モジュール592のセットによりモジュラ・コントローラ・メインフレーム500と相互結合されたデバイス制御ドメインのランプユニットを制御する。

【0197】現代の照明システム用の制御装置は、異なる通信プロトコルと、機能と、データ要件を有する別種のランプユニットを同時にサポートすることができなければならないことがしばしばある。例えば、照明デザイナーは、自動化可変パラメータ・ランプユニットの他に、(光度のみが)慣用のランプユニットを組み入れようとし、同様に、異なる製造業者により提供されるランプユニットを利用しようとするのがしばしばある。

【0198】慣用のランプユニットは光度データ値のみを必要とするが、自動化可変パラメータ・ランプユニット、例えば、Vari*Lite (商標登録) Series 200 (商標) 照明システムのようなランプユニットは、色と、高度と、遮光板(gobo)と、パンと、チルトとを含む数多くの可変パラメータを必要とすることになる。

【0199】その上、イメージをプロジェクトすることができる複雑な自動化可変パラメータ・ランプユニット、例えば、出願07/693,366号に開示されているようなLCD(liquid crystal display)を有するランプユニットは、典型的な自動化ランプユニットと関係付けをしたパラメータ・データの他に、イメージデータ・ファイルを必要とする。従って、各モジュールは、次に詳細に説明する入力および出力モジュール590、592のセット内で、独立のデータ・ネットワークとして構成されている。また、各モジュールは、各モジュールに接続された具体的なデバイスと通信するため、1つ以上の異なる通信プロトコルに従うことができる。よって、モジュラ・コントローラ・メインフレーム500を数多くの互換性のないデバイスに接続することが容易になる。このように、モジュラ・コントローラ・メインフレーム500は、別種の通信プロトコルを有する複数の制御デバイスと、複数のランプユニットと、別種の通信プロトコルと機能とデータフォーマットとを有する他の出力デバイスとの間で通信を行うためのインタフェース・システムとしてサージする。従って、別種の機能とプロトコルを有する種々のタイプのランプユニットを制御することがで

きる。

【0200】図26および図27に示す入力モジュール590のセットのモジュールは、次に詳細に説明するが、それぞれ、モジュラ・コントローラ・メインフレーム500と、このメインフレーム500に接続されている1つ以上の制御デバイス、例えば、556-558, 560, 568, 570とのインタフェースとしてサーブする。入力モジュール590のセットのモジュールは、それぞれ、入力モジュール590に接続されている制御デバイスからの制御コマンドを受信し、必要な場合には、モジュラ・コントローラ・メインフレーム500が解釈する(interpret)ことができるコマンド・フォーマットに、受信されたコマンドを変換するように、構成されている。

【0201】入力モジュール592のセットのモジュールは、次に詳細に説明するが、それぞれ、モジュラ・コントローラ・メインフレーム500と、1タイプ以上のランプユニットと、別種の通信プロトコルと機能とデータ・フォーマットを有する他の出力デバイスとのインタフェースとしてサーブする。出力モジュール592のセットの個々のモジュールは、モジュラ・コントローラ・メインフレーム500からのジェネリック・コンソール・コマンドであって、全ランプユニットおよび他の出力デバイスとの通信に有用なコンソール・コマンドを受信し、必要な場合、受信されたジェネリック・コンソール・コマンドを、出力モジュール592に接続されている具体的なタイプの出力デバイスと通信を行うのに必要な具体的なコマンドまたはパラメタに変換する。

【0202】制御パネル・ユニット546-552の調整つまみ、例えば、次に詳細に説明する、ノブと、ボタンと、フェーダとを操作すると、モジュラ・コントローラ・メインフレーム500は、各ランプユニットに伝送するため、コンソール調整つまみの操作を符号化表現したものよりなる「ジェネリック」コンソール・コマンドを生成するのが好ましい。

【0203】ローカル・プロセッサと、キュー・データをストアするためのメモリとを有する幾つかのランプユニット、例えば、上述したランプユニットは、これらのコンソール・コマンドを直接獲得し、ランプユニット・プロセッサが、それらのパラメタを一致させて所望の照明効果を得るのに必要な値に変換することができ、一方、このような高い処理能力を持たない他のランプユニットは、絶対パラメタ・データ、すなわち、パンと、チルトと、色と、他のパラメタに対する実際に計算されたデータ値のみを獲得することができる。

【0204】従って、実際のパラメタ・データのみを獲得するこれらのランプユニットには、それらのランプユニットに最終的に伝送する前に、それらのランプユニットからリモートのプロセッサにより、「ジェネリック」コンソール・コマンドを、必要な具体的な絶対パラメタ

・データ値に変換しなければならない。好ましくは、これらジェネリック・コンソール・コマンドは、出力モジュール592のセットのモジュールにそれぞれ伝送され、必要な場合には、出力モジュール592は、それらのコマンドをランプユニットに伝送する前に、コマンド変換を行う。

【0205】例えば、マニュアル制御のために選択された数多くのランプユニットに対して、光度制御に関係付けをしたノブをコンソール・オペレータが調整する場合、モジュラ・コントローラ・メインフレーム500は、コンソール・オペレータによるノブ調整に対応する「デルタ値」の符号化表現よりなるコマンドを、出力モジュール592のセットのモジュールにそれぞれ伝送することになる。出力モジュール592のセットのうち、このジェネリック・コンソール・コマンドを直接解釈することができるランプユニットをサポートするモジュールは、このコンソール・コマンドを、接続された各ランプユニットに変換せずに、伝送することになる。しかし、出力モジュール592のセットのうち、絶対パラメタ・コマンドを必要とするランプユニットをサポートするモジュールは、選択されたランプユニットを所望の光度に設定するのに必要な絶対パラメタ値に、これらジェネリック・コンソール・コマンドを変換しなければならない。

【0206】出力モジュール592のセットのうち、コマンド変換するように構成されているモジュールは、照明制御装置の機能、例えば、米国特許第4,392,187号

(発明者: Bornhorst) に記載されている照明制御装置の機能を複製するため、プロセッサと、キュー・データをストアするためのメモリとを含む。米国特許4,392,187号には、照明制御装置はコンソール調整つまみの操作を表すコマンドを受信し、各ランプユニットに伝送するのに必要な絶対パラメタ値を計算することが開示されている。さらに、接続されたランプユニット用の適正な通信プロトコルに従って、通信された信号を一致させるように、出力モジュール592をそれぞれ構成することができる。すなわち、ランプユニットへは、適正な信号レベルと、タイミングと、パラメタ順番と、当該ランプユニットにより予測される他のフォーマット・ファクタが伝送されることを、出力モジュールがそれぞれ保証する。

【0207】モジュラ・コントローラ・メインフレーム/メイン・プロセッサ・カーネルモジュラ・コントローラ・メインフレーム500は、図26に示すように、メイン・プロセッサ・カーネル502と、入出力モジュール590, 592を含む。入出力モジュール590, 592は高速パラレル・データ・バス(入力バス512と、出力バス572と、大容量記憶バス504を含む)により相互結合されている。

【0208】メイン・プロセッサ・カーネル502はMo

torola MC68040のようなマイクロプロセッサ (CPU) と、RAM(random-access memory)と、ROM(read-only memory)と、関係付けをしたサポート回路とを含む。メイン・プロセッサ・カーネル502は、CPUと、メモリ (RAMおよびROM) とを有するマザーボードでコンストラクトすることができる。このマザーボードはその他に、種々のバス504, 512, 572のプラグイン・コネクタと一致させる(mate with) させるため、ビルトイン・コネクタを有する。

【0209】メイン・プロセッサ・カーネル502は、幾つかのモードのうちのいずれかのモードで、入力および出力モジュール590, 592のセットのうちの種々のモジュールと通信をおこなって、キューデータと、ステータス・レポートと、他の情報を転送する。コンソールと個々のランプユニットとで通信を行なう上述した通信管理プログラムと同様の方法で、メイン・プロセッサ・カーネル502と、入力および出力モジュール590, 592のセットのうちの種々のモジュールとで通信するための2タイプのメッセージ・アドレス、すなわち、個々のモジュール・アドレスとモジュール・ブロードキャスト・アドレスとを提供することができる。したがって、メイン・プロセッサ・カーネル502は、当該特定のモジュールと関係付けをした一意のモジュール・アドレスを伝送して、各モジュールに個別にアクセスすることができる。このようにすると、モジュールは全て個別のモジュール・アドレスとの伝送を受信することができるが、伝送されたアドレスと関係付けをしたモジュールだけが応答することができる。

【0210】モジュール・ブロードキャスト・モードでは、メイン・プロセッサ・カーネル502は入力および出力モジュール590, 592のセットのうち、共通のアドレスの全てのモジュールにメッセージを伝送することができる。メイン・プロセッサ・カーネル502は入力および出力モジュール590, 592のセットのモジュールは、それぞれ、モジュールの個々の構成に依存するとともに、当該モジュールに接続されているデバイスに依存して応答する。1つの実施例では、メイン・プロセッサ・カーネル502がブロードキャストを入力モジュール590のセットまたは出力モジュール592のセットのいずれか一方に限定するためのモジュール・ブロードキャスト・アドレスを追加することができる。

【0211】ランプユニット・ブロードキャスト・モードでは、モジュラ・コントローラ・メインフレーム500により受信された、任意の制御入力デバイスからの制御入力信号は、出力モジュール592のセットにより、デバイス制御ドメインで接続されているランプユニットのうち、共通のアドレスのランプユニットにそれぞれ伝送される。

【0212】大容量記憶デバイス
メイン・プロセッサ・カーネル502は大容量記憶バス

504を介してプロセッサにより制御される大容量記憶および検索デバイス、例えば、大容量ハード・ディスク・ドライブ506と、デジタルデータ・テープカートリッジ・ドライブ508と通信を行なう。さらに、フロッピー・ディスク・ドライブ510を直接にメイン・プロセッサ・カーネルに接続させることができる。

【0213】大容量記憶バス504はSCSI (Small Computer System Interface) プロトコルと一致させるか、あるいは、同様のバス標準と一致させ、図示しない追加の大容量記憶デバイスを大容量記憶バス504に容易に接続させることができるのが好ましい。

【0214】ランプユニットからのキューデータ・アップロードであって、出力モジュール592により受信され、メイン・プロセッサ・カーネル502に伝送されたキューデータ・アップロードを、ハードディスク・ドライブ506にストアすることができ、さらに、テープ・カートリッジ・ドライブ508および/またはフロッピーディスク・ドライブ510に「バックアップ」することができる。メイン・プロセッサ・カーネル502か、あるいは、パーソナル・コンピュータ560 (図27) に対してオペレートするデータベースおよびレポート・ジェネレータ・プログラムにより分析するため、ステータス・レポートをハードディスク・ドライブ506にログ(log) することができる。ステータス・レポートは次に詳細に説明する。オペレーティングシステム
モジュラ制御システムの種々のデバイス、例えば、メイン・プロセッサ・カーネル502と、ランプユニットと、スマート・リピータ52-58のためのオペレーティングシステム・ソフトウェアを、ハードディスク・ドライブ506にストアすることができる。このようにして、オペレーティングシステム・ソフトウェアの最新バージョンを、テープドライブ508に挿入されたテープ・カートリッジか、あるいは、フロッピーディスク・ドライブ510に挿入されたフロッピーディスクを介して、ハードディスク・ドライブにロードすることができる。その後、オペレーティングシステム・ソフトウェアを、必要に応じて、大容量記憶デバイス506-510から適正なデバイスにダウンロードすることができる。入力モジュールマイクロコントローラ集積回路、例えば、Motorola MC68302 Integrated Multiprotocol Processorのような集積回路によりそれぞれ制御される入力モジュール590のセットに、上述したように、メイン・プロセッサ・カーネル502が接続されている。例えば、16bit または32bit データバスであって、必要に応じて、関係付けをしたアドレスおよび制御ラインを有するパラレル入力バス512により、メイン・プロセッサ・カーネルに、入力モジュール590のセットが接続される。

【0215】図26に示す入力モジュール590の構成は単に説明のためだけに過ぎない。この入力モジュール

を次に説明する。将来開発することができる、新しい制御デバイスとシリアル・リンク・フォーマットに適合させるため、必要に応じて、他の入力モジュールを入力バス512に接続させることができる。

【0216】入力モジュール514は制御パネルユニット546-552を、図27に示すように、モジュラ・コントローラ・メインフレーム500に、周知のシリアル・トークンバス523か、あるいは、同様のシリアルデータ・リンクを介して接続する。マイクロプロセッサ制御コントロールパネルユニット546-552と通信を行なうため、入力モジュール514をトークンバス制御装置として具現化することができる。シリアル・トークンバスに接続されたマイクロプロセッサ・ステーションは、制御パケット、すなわち、「トークン」を所持したとき、伝送制御が制御パケットを受信したプロセッサに移り、このマイクロプロセッサ・ステーションはこのシリアル・トークンバスに伝送することができる。あるいはまた、種々のデバイスをデジタイズチェインで接続して閉じたリングを形成し、トークンリングをインプリメントすることができる。

【0217】入力モジュール514はこのシリアル・トークンバス上の全メッセージを「リッスン(listen)」するように構成するのが好ましい。このようにして、バス512に接続されているどのプロセッサ・ステーションがこの制御パケットを所持するかに関らず、入力モジュール514はいつでも入力制御信号を(適正な)メイン・プロセッサ・カーネル502に伝送することができる。

【0218】制御パネルユニット546-552は、それぞれ、1つ以上のマイクロプロセッサにより制御され、特定の機構と、上述した制御コンソールの機能、例えば、Vari-Lite, Inc., of Dallas, Texas が販売するArtisan(登録商標)コンソールのような制御コンソール機能を、補助の機構の他に、組み入れることができる。

【0219】例えば、マニュアル制御パネル・ユニット548は、例えば、マニュアル制御されるランプユニットを選択するための手段と、選択されるかあるいはアクティブなランプユニットを示すための手段と、選択されたランプユニットの種々のパラメタを操作する手段とを提供する。典型的な照明構成では、「cue」として得られる照明効果をストアする前に、所望の構成で種々のランプユニットを構成するために、マニュアル制御が利用される。

【0220】さらに、サブマスタ制御パネル550は、例えば、キューをストアし、リコールし、開始する手段を提供し、同様に、2つのキューの間でマニュアル・クロス・フェードを行なう手段を提供する。一度、キューがサブマスタに割り当てられると、関係付けをしたサブマスタを選択し、そのサブマスタのフェーダを操作して、そのサブマスタを活動化することができるのが典型

的である。

【0221】チェース/マトリクス制御パネル・ユニット552は、ランプユニットの小グループを制御するマトリクス制御手段を提供し、さらに、チェース・シーケンスを制御する手段を提供することができる。チェースはキューのプログラムされたシーケンスであり、チェースをチェース/マトリクス制御パネル・ユニット552にリコールすることができ、公知の方法で自動的に実行することができる。光源がマトリクス・パネルに予め「パッチ(patch)」されてる場合は、リコールされたキューで、光源の光度を操作するため、公知のマトリクス・パネルにはフェーダが用意されている。

【0222】追加の制御パネル・ユニット546は、マニュアル調整つまみのサブセットと、キュー記憶およびリコールと、上述したチェースおよびマトリクス調整つまみがあり、この制御パネル・ユニット546を、リモート・ロケーションの代替制御コンソールとして用いることができるか、あるいは、バックアップ制御コンソールとして用いることができる。リモートまたはバックアップ制御コンソール、例えば、Vari-Lite, Inc., of Dallas, Texas が販売するmini-Artisans(登録商標)コンソールのような制御コンソールの特定の機構および機能を、補助機構の他に組み入れるように、このリモートまたはバックアップ・コンソールを構成することができる。制御パネル・ユニット546を単一の制御パネル上にコンストラクトするのが好ましい。

【0223】シリアル・トークンバス523に接続されている制御パネル・ユニット546-552が、それぞれ同時にオンラインすることができるだけのデータ・レートで、データを受信しサンプリングするように、入力モジュール514を構成するのが好ましい。このように構成すると、リモートのオペレータは、リモート制御パネル・ユニット546に対して制御オペレーションを行なうことができ、他方、メイン・オペレータは1つ以上の制御パネル・ユニット548-552に対してオペレーションを行なう。制御パネル・ユニット546-552は1つ以上のディスプレイ・モジュールを含むのが好ましい。ディスプレイ・モジュールとしては、LCD(liquid crystal displays)か、EL(electro-luminescent)グラフィック・ディスプレイ・パネルか、VF(vacuum-fluorescent)英数字ディスプレイ・モジュールか、LED(light emitting diode)キャラクタ表示素子か、あるいは、他の適正なディスプレイ・デバイスでもよい。

【0224】制御パネル・ユニット546-552も「ソフト・スイッチ」を含むことができる。ソフト・スイッチとしては、例えば、プッシュボタンのキーキャップにキャラクタ表示手段を有するプッシュボタン、例えば、Industrial Electronic Engineers, Inc., of Van Nuys, California製のPixie Graphic LCD Switchのよう

なプッシュボタンでも良い。あるいはまた、プッシュボタンを、このプッシュ・ボタンの機能を表示する他のディスプレイ・ユニットに近接させてマウントすることができる。

【0225】このようにすると、プッシュボタンの機能が異なるオペレーティング・モードで変化するか、あるいは、異なるオペレーティングシステム・ソフトウェア・バージョンの制御で変化する時、プッシュボタンのキーキャップに表示されるラベルか、あるいは、近接するディスプレイ・ユニットに表示されるラベルを、書き換えて、変更された機能が、あるいは、代替機能を表示することができる。

【0226】プッシュボタン・スイッチの中には、例えば、数字キーパッドのように、機能を固定することができるものがある。一方、オペレーションの異なるモードで、他のボタンを再プログラムすることができる。例えば、ボタンのバンクが、あるモードでは、チャンネル・セレクト・ボタンになり、あるモードでは、タイミング機能調整つまみになることができる。

【0227】制御パネル・ユニット546-552も、1つ以上のタイプの連続マニュアル制御デバイス、例えば、ロータリ光学エンコーダのようなロータリノブ・タイプの制御装置か、あるいは、リニア・スライド・ポテンシオメータのようなリニア・フェーダ・タイプの制御装置を含むことができる。マニュアル制御パネル548はロータリ・コンティニユアス制御装置のみを含むことができ、サブマスタ制御パネル550はリニア・コンティニユアス制御装置のみを含むことができる。特殊機能制御パネル、例えば、チェース/マトリクス制御装置552またはリモート/バックアップ・コンソール546のような制御パネルは、ロータリ制御装置とリニア制御装置の両方の制御装置を含むことができる。

【0228】1次制御パネル・ユニット548-552を、メイン制御コンソールとしてサブするため、共に、単一のコンソールにマウントすることができる。あるいはまた、1次制御パネル・ユニット548-552を、別々に、離れたロケーションにマウントすることができる。さらに、制御パネル・ユニット546-552を特別なアプリケーションのために、ノブの所望のミックスと、フェーダと、ディスプレイ・ユニットをインストールし、希望に応じて、ソフト・スイッチをプログラミングして、カスタマイズすることができる。

【0229】制御パネル・ユニット546-552をモジュラ設計することにより、照明システム・オペレータの圧迫的な要望に従って、その制御装置を設計することができる。そして、それらの要望が変わった場合、照明制御コンソール全体のハードウェアとオペレーティングシステム・プログラムを完全に設計し直すことなく、その制御装置を異なる構成にすることができる。制御パネル・ユニット546-552は自分のオペレーティング

システム・プログラムを稼働させるので、新しい制御パネル・ユニットをサポートするため、必要なプログラム・モジュールを書き直すことができ、一方、モジュラ制御コンソール・システムの残りは変更しない。

【0230】MIDI (Musical Instrument Digital Interface) モジュール516か、Ethernetポート・モジュール518か、RS232 シリアルデータ・ポート・モジュール520か、あるいは、ビデオ入力モジュール522を含めて、同様に、追加の入力モジュールを、図26に示すように提供することができる。

【0231】MIDI入力モジュール516は、MIDIコンベンションに従った信号を受信し翻訳するように構成されている。第1の実施例では、モジュラ・コントローラ・メインフレーム500により生成された制御コンソール・コマンドを記録し再生するため、図27に示すように、シリアルデータ・リンク524により、Alesis MMT-8 Multi Track MIDI RecorderのようなMIDIレコーダ/シーケンサ556に、MIDI入力モジュール516を接続することができる。詳細は次に説明する。

【0232】第2の実施例では、電子楽器557か、あるいは、MIDI「ノート(note)」を生成することができる他のデバイスに、MIDI入力モジュール516を接続することができる。この他のデバイスにより生成されたMIDIノートは、照明パラメタを効率的に制御するため、モジュラ・コントローラ・メインフレーム500により受信され翻訳される。図27に示すように、電子楽器557をMIDIレコーダ/シーケンサ556を介してMIDI入力モジュール516に接続することができる。あるいはまた、電子楽器557をMIDI入力モジュール516に直接結合することができる。

【0233】MIDIコンベンションは128個のMIDIノートを定義しているので、種々の制御パネル・ユニット546-552上の最高128個のボタンを、一意のMIDIノートに「マッピング」することができる。従って、制御パネル・ユニット546-552上のマッピングされたボタンを押下すると、メイン・プロセッサ・カーネル502によりコンソール・コマンドが検出され、そして、メイン・プロセッサ・カーネル502はそのコンソール・コマンドを「マッピング」されたMIDIコマンドに変換することができる。そして、MIDIレコーダ556により記録するため、MIDI NOTE ON コマンドをMIDI入力モジュール516に伝送することができる。同様に、当該ボタンの押下が解除されると、MIDI NOTE OFF コマンドをレコーダ556に伝送することができる。

【0234】MIDI IN ポートで受信されたイベントを記録するようにレコーダ556を構成することにより、制御パネル・ユニット546-552上のボタンを押下して生成された一連のコマンドを、後程取り出すため、レ

コード556の不揮発性メモリにストアすることができる。

【0235】このようにして、特定の歌または踊りに対して一連のキューを実行するため、制御パネル・ユニット546-552を特別の条件にする「Board Control Cues (ボード制御キュー)」を確立することができる。例えば、特定の歌に対してサブマスタ制御パネル550を用意するため、当該歌または踊りに対して関係付けをしたキュー番号を具体的なサブマスタにロードし、ついで、適正なサブマスタを選択しなければならない。同様に、チェース/マトリクス制御パネル552を特定の歌に対して用意するため、あるチェース/マトリクス・パッチをロードしなければならない。

【0236】このコンソールのオペレータは、これらのタスクを、演技の間、それぞれの歌の前に、この制御パネルの一連のボタンを押下して行うのが典型的である。運悪く、歌われる歌の順番が前もって分からないことがよくあるが、制御コンソールのオペレータは、次に歌われる歌の知らせに従って、制御パネル546-552を必要な構成にするだけの最小の時間しか持たないことになる。

【0237】従って、MIDIレコード556は歌や踊りにそれぞれ関係付けをしたキューのシリーズに対する特定の条件に、制御パネル・ユニット546-552をするのに必要な一連のボタン押下を、(前もって)それぞれの歌に対して記録することができる。その後、演技の間、このコンソールのオペレータに、次に歌われる歌が知らされると、前もって記録されたボタン押下のプレイバックを、MIDIレコード556により開始することができる。制御パネル・ユニット546-552を特定の条件にセットアップするのに必要なキー押下のクイック・パーストを、MIDIレコード556は効率的に行うことができる。

【0238】MIDIレコード556からのMIDI信号であって、MIDI入力モジュール516で受信されたMIDI信号が、プレイバックの間、MIDI入力モジュール516のプロセッサにより、対応する制御コマンドに変換してから、メインプロセッサ・カーネル502に伝送する(その後、制御パネル・ユニット546-552に伝送するため)のが好ましい。MIDI入力モジュール516で、対応する制御コマンドへの変換が行われるので、メインプロセッサ・カーネル502のプロセッサは、このタスクを行うことから解放され、メインプロセッサ・カーネル502は自由に他のタスクを行うことができる。

【0239】上述したように、MIDIレコード556を、1つ以上のトラックを同時にプレイすることができるマルチトラックMIDIレコードとして具現化することができるが、NOTE ON コマンドとNOTE OFFコマンドが同時に同一ボタンに向けられないように記録およびプレ

イバックするため、シングル・トラックのみを利用すべきである。さらに、制御パネル・ユニット546-552のマッピングされていないボタンを押下しても、MIDIレコード556により記録されないことに、注意すべきである。

【0240】代替の実施例では、電子楽器か、あるいは、MIDIノートを生成することができる他のデバイスにより生成された光源パラメタ制御コマンドを受信するように、MIDI入力モジュール516を構成することができる。上述したように、MIDIノートをパラメタ制御コマンドにマッピングすることにより、MIDI楽器による所望の「ノート」に対応する「キー」を押下して、照明システム・パラメタをリモート・ソースから直接変更することができる。

【0241】MIDIデバイス557により生成されたMIDI NOTE ONコマンドおよびMIDI NOTE OFF コマンドは、MIDI入力モジュール516により照明システムに通信されることになる。MIDI入力デバイス557では、メインプロセッサ・カーネル502に伝送する前に、MIDI入力モジュール516のプロセッサにより、そのMIDIコマンドを対応するコンソール制御コマンドに変換するのが好ましい。

【0242】モジュラ・コントローラ・メインフレーム500で受信されたコマンドであって、コンソール・ボタンのステートを制御するためのコマンドを、2つの方法、すなわち、コンソール・ボタンのステートをその代替ステートにトグルするか、あるいは、コンソール・ボタンのステートを、前の状態に関係なく、所望のステートにするかのいずれかの方法で解釈することができる。従って、2つのモードを提供するのが好ましい。第1のモードでは、モジュラ・コントローラ・メインフレーム500により受信されたパラメタ制御コマンド、すなわち、MIDI NOTE ONコマンドまたはMIDI NOTE OFF コマンドを、関係付けをしたコンソール・ボタンをその代替ステートにトグルするコマンドと解釈することになる。

【0243】しかし、第2のモードでは、モジュラ・コントローラ・メインフレーム500により受信されたコマンドは、関係付けをしたボタンを、コンソール・ボタンの前のステートに関係なく、公知のステートにするコマンドと解釈される。

【0244】例えば、入力モジュール590のセットの1つのモジュールにより受信されたメッセージは、そのメッセージが入力モジュールにより受信されたとき、サブマスタが現在選択されたか否かに関らず、あるサブマスタを選択するか、および/あるいは、他のサブマスタを選択解除するコマンドを含むことができる。

【0245】モジュラ・コントローラ・メインフレーム500によりこのコマンドが受信されると直ちに、メインプロセッサ・カーネル502はあるサブマスタを選択および/または選択解除するコマンドを識別する。メイ

ンプロセッサ・カーネル502は、このメッセージを制御パネル・ユニット546-552に向け、その後、そのコマンドを含むメッセージを、制御パネル・ユニット546-552に伝送するため、入力モジュール514に送信する。

【0246】制御パネル・ユニット546-552は、それぞれ、そのメッセージを受信し、指定されたサブマスタが個々の制御パネル・ユニットに駐在するか否かにより、個々に応答することになる。その指定されたサブマスタを有する制御パネル・ユニットは、適正な選択ボタン[SEL]を、まだ点灯していない場合には、関係付けをしたサブマスタが選択されたことを示すため、点灯させて、そのコマンドに応答することになる。指定されたサブマスタがない制御パネル・ユニットは、サブマスタ・セレクト・コマンドを無視することになる。

【0247】その上、サブマスタ選択/選択解除コマンドを含むメッセージも、適正な場合に、各ランプユニットに送信するため、メインプロセッサ・カーネル502により、セット592の種々の出力モジュールにそれぞれ送信すべきである。上述したように、「ジェネリック」コンソール・コマンドを解釈することができないランプユニットに対して、関係付けをした出力モジュール592はそのコマンドを、ランプユニットが解釈するのに適正な絶対パラメータデータに変換しなければならない。

【0248】図26に示す入力モジュール518は、高速度データ通信用のEthernetポートをインプリメントする。インプリメントの一例では、Ethernetポートを、Sun Microsystems SPARC 2コンピュータ・ワークステーションのようなグラフィックス・ワークステーションに、図27に示すように、シリアル・データリンクを介して接続することができる。グラフィックス・ワークステーション558を利用して、イメージ生成能力を有する自動化ランプユニットによりプロジェクトされるイメージを展開し、修正し、制御することができる。このような自動化ランプユニットとしては、例えば、出願07/693,366号で開示されている液晶プロジェクション・ゲートを有するランプユニットがある。

【0249】その上、グラフィックス・ワークステーション558を利用して、照明パラメータのオフライン・プログラミングをコーディネートするのに適したソフトウェアをオペレートすることができる。このコーディネートは、演技開催場所の3次元モデルと、照明システムの機能、例えば、出願07/641,031号で開示されたプログラミングおよびモデリング・ツールを利用して行われる。

【0250】入力モジュール520はRS232C互換シリアルデータ・ポートをインプリメントするのが好ましい。一実施例では、図27に示すように、RS232ポートをシリアルデータ・リンク528を介してパーソナル・コンピュータ560に接続することができ、キューデータ

と、ステータス・レポートと、他の情報を、種々のランプユニットと、制御パネル・ユニット546-552と、パーソナル・コンピュータ560に転送することができる。このようにして、パーソナル・コンピュータ560を、キューデータおよびステータス・レポートを展開し表示し操作するために、利用することができる。

【0251】適正なモデム回路に、入力モジュール520が含まれることが好ましい。というのは、RS232データ・フォーマットを電話回線を介して、普通、伝送するからである。よって、図27に示すように、モジュラ・コントローラ・メインフレーム500は、リモート・デバイスおよびコンピュータ・システムと、シリアル・データ・リンク530および電話回線インタフェース562を介して通信することができる。その結果、キューデータと、ステータス・レポートと、他の情報を、演技開催場所とリモート保守機構との間で、伝送することができる。このようにすると、照明システムのフォールトを、リモート保守機構により診断することができる。

【0252】図27に示すように、アナログ・ビデオ入力を、例えば、ビデオテープ・レコーダ570および/またはビデオカメラから、シリアルデータ・リンク532を介して受信するように、入力モジュール522を構成するのが好ましい。このようにすると、ビデオテープ・レコーダ570および/またはカメラ568により生成されたビデオ信号を、マルチプレクシングし、イメージ生成能力を有するランプユニットに、個別のデータリンク584（上述した）を介して、伝送することができる。

【0253】出力モジュール上述したように、メインプロセッサ・カーネル502とランプユニットをインタフェースするため、出力モジュール592を提供するのが好ましい。出力モジュール592のセットは、メインプロセッサ・カーネル502に、パラレル出力バス572、例えば、必要なアドレス線およびコントロール線の他に、32bitータバスまたは64bitデータバスを有するパラレル出力バスにより、接続することができる。

【0254】セット592の各出力モジュールは、1つ以上のマイクロプロセッサ、例えば、Motorola MC68302 Integrated Multiprotocol Processor およびMotorola MC68332 microcontrollerにより制御されるのが好ましい。

【0255】上述したように、出力モジュール592のセットのモジュールは、それぞれ、次のような独立のデータネットワークとして構成されるのが好ましい。すなわち、各モジュールが、モジュラ・コントローラ・メインフレーム500と、1タイプ以上のランプユニットと、別種の通信プロトコルと機能とデータフォーマットを有する他の出力デバイスとのインタフェースとしてサーブすることができるネットワークとして構成されるのが好ましい。出力モジュール592のセットの出力モジ

ジュールは、それぞれ、必要な場合、モジュラ・コントローラ・メインフレーム500から受信されたジェネリック・コンソール・コマンドを、具体的なタイプのランプユニットまたは当該ランプユニットに接続された出力デバイスと通信をおこなうのに必要な具体的なコマンドまたはパラメタに変換する。

【0256】次に説明するが、図26および図27に示すように、出力モジュール592のセットのモジュールの構成と、ランプユニットまたは当該ランプユニットに接続された出力デバイスの構成と、例えば、各出力モジュールに接続されたランプユニットの数とタイプは、単に説明のためであり、ランプユニットの種類と数と構成の種々の選択(preference)に適合させ、各照明システムを備えた他の出力デバイスに適合させるため、必要に応じて、他の出力モジュール構成を容易に開発することができる。さらに、ランプユニットおよび出力デバイスのために開発することができる、新しい通信プロトコルか、機能か、データ・フォーマットに適合させるため、必要に応じて、追加の個別の出力モジュールをコンストラクトし、グレードアップすることができる。

【0257】上述したように、セット592の各出力モジュールを、接続されたランプユニットに伝送する前に、適正な通信プロトコルと、信号を一致させることができないなければならない。すなわち、ランプユニットへの伝送では、適正な信号レベルと、タイミングと、パラメタ順序と、ランプユニットにより予測される他のフォーマット・ファクタを有することを、各出力モジュールは保証しなければならない。

【0258】本実施例の照明システムは、ジェネリック・コンソール・コマンド、例えば、Vari*Lite (登録商標) Series 200 (商標) lighting system (VL200S) 例えば、VL2 (登録商標) か、VL2B (登録商標) か、VL3 (商標) か、あるいはVL4 (商標) luminairesのような照明システムと関係付けをしたジェネリック・コンソール・コマンドを直接受信することができる自動化可変パラメタ・ランプユニットと、絶対パラメタ・コマンド、すなわち、ジェネリック・コンソール・コマンドを変換するのに必要なコマンドを獲得する自動化可変ランプユニットと、ストアされているイメージをプロジェクトすることができる自動化可変ランプユニットと、ビデオ・イメージをプロジェクトすることができる自動化可変ランプユニットと、標準慣用固定焦点ランプユニットおよび他の出力デバイス、例えば、エアー・キャノンと、特殊降下プロジェクトと、スモーク・マシーンと、トラス(truss) アセンブリのエリベーションを制御するためのチェーンホイスト・モータと、よりなる。

【0259】好ましくは、上述したように、出力モジュール592のセットの各モジュールであって、受信されたコマンドに応答することができるランプユニットをサポートするモジュールは、コマンドを共通ランプユニッ

トにブロードキャストすることができ、ランプ固有のコマンドを個々のランプユニット・アドレスに伝送する。

【0260】その上、出力モジュール592のセットは、大容量記憶デバイス506-510からのオペレーティングシステム・ソフトウェアを、個々のランプユニットと、これらランプユニットに接続されたリピータにダウンロードするように構成するのが好ましい。

【0261】出力モジュール592のセットのモジュールを、それぞれ、同様に、当該出力モジュールのために書かれたオペレーティングシステム・ソフトウェアに従って、他のタスクを遂行するようにプログラムすることができる。

【0262】本実施例では、出力モジュール534を、ジェネリック・コンソール・コマンドを直接受信することができる自動化可変ランプユニット、例えば、Vari*Lite (登録商標) Series 200 (商標) systemと関係付けをしたランプユニットをサポートするように構成することができる。出力モジュール534がVari*Lite (登録商標) Series 200 (商標) systemの通信プロトコルか、あるいは、同様のプロトコルをサポートするように構成されたときは、Manchesterエンコーディングを用いた両方向シリアル・データリンク26A2であって、デュアル・ネットワークで、個別のブロードキャストと、応答データリンクをフィーチャするデータリンクをインプリメントするのが好ましい。

【0263】絶対パラメタ値を必要とする自動化可変ランプユニットをサポートするように、出力モジュール535が構成されている。従って、出力モジュール535はメインプロセッサ・カーネル502から受信されたジェネリック・コンソール・コマンドを、出力モジュールに接続されたランプユニットの具体的なタイプに対して必要とする絶対パラメタ値に変換することができない。上述したように、出力モジュール535は、必要なコマンド変換を行うため、プロセッサと、キューデータをストアするためのメモリとを含むのが好ましい。

【0264】本実施例では、イメージをプロジェクトすることができる自動化可変ランプユニット、例えば、液晶プロジェクト能力を有するランプユニットをサポートするように、出力モジュール536を構成するのが好ましい。従って、演技の前に、ランプユニットのローカルメモリにストアするため、典型的には、コントローラによりダウンロードされたシステム・コマンドおよびパラメタ・データで、ストアされているデジタル・データ・ファイルをインタリーブするように、データリンク26A4を介して、関係付けをしたランプユニットに接続した出力モジュール536を構成するのが好ましい。このイメージ・データ・ファイルを、グラフィック・ワークステーション558から受信するか、あるいは、大容量記憶デバイス506-510から受信することができ

る。

【0265】出力モジュール536に関係付けをしたこのイメージ信号はデジタル・イメージ信号であるのが典型的である。従って、これらの信号を、出力モジュール536とリピータ54A4の間で搬送して、接続されたランプユニットに、典型的には照明システムで利用される専用のツイストペア574により伝送することができる。

【0266】ビデオイメージをプロジェクトすることができる自動化ランプユニット、例えば、液晶プロジェクション能力を有するランプユニットをサポートするように、出力モジュール538を構成することができる。従って、出力モジュール538は、データリンク26A5をインプリメントして、アナログ・ビデオ信号を、接続されているランプユニットに伝送することができる。アナログ・ビデオソースを接続している入力モジュール522、例えば、上述したように、カメラ568またはビデオテープレコーダ570を介して、ビデオ信号を受信することができる。

【0267】出力モジュール538に関係付けをしたビデオ信号は、アナログ信号であるのが典型的であるので、ビデオ信号を、出力モジュール538と、接続されたランプユニットの間で、専用の同軸ケーブル576により搬送することができる。デジタル・ビデオ信号を適合させるのに必要な時間で、出力モジュール538を再構成することができる。

【0268】ストアされたデジタル・データファイルを逐次化して、アニメートされたピクチャを生成するように、液晶プロジェクション能力を有するランプユニットを構成することができる。従って、イメージ・ファイルをこれらランプユニットにリアルタイム・シーケンスで伝送するように、出力モジュール536および538が構成されるのが好ましい。あるいはまた、イメージ・データファイルを演技の前にストアするため、ランプユニットに伝送することができ、キューデータを取り出すのと同様の方法で、シーケンシャルにリコールすることができる。アナログ・ビデオ信号を獲得するこれらビデオ・ランプユニットに対して、シーケンシャルにプロジェクトするためのデジタル・イメージ・データファイルを、ランプユニットに伝送する前に、出力モジュール538によりアナログ信号に変換するのが好ましい。

【0269】イメージ生成ランプユニットをサポートする出力モジュール536および538は、イメージ処理するための適正なハードウェア、すなわち、イメージファイルをストアするための追加のメモリと、必要な場合には、これら出力モジュールに接続されるランプユニットのタイプに応じて、イメージファイルをアナログ信号からデジタル信号に変換するか、あるいは、デジタル信号をアナログ信号に変換する回路を含むのが好ましい。その上、虞らく、これら出力モジュールは、イメージデ

ータ・ファイルを伝送する高い伝送能力、すなわち、高速光ファイバ・データリンクを必要とするか、あるいはまた、比較的低速のデータリンクを介して、アニメーションのシーケンスをリアルタイムに伝送するための圧縮回路を必要とする。

【0270】図26に示す実施例では、個別のデータリンク584を、ビデオ入力モジュール522とビデオ出力モジュール538の間に設け、その間で、ビデオ信号が伝送される。アナログ・ビデオ信号の場合は、モジュール522でアナログ→デジタル変換を行い、ついで、モジュール538でデジタル→アナログ変換を行わずに、追加のリンク584により、ビデオ信号を入力モジュール522から出力モジュール538に、アナログ信号として伝送することができる。メインプロセッサ・カーネル502と内部バス512および572の伝送能力を過負荷状態にしないようにするため、デジタル・ビデオ信号が利用される場合でさえ、追加リンク584が利用されるのが好ましい。

【0271】図27に示すように、幾つかのビデオ入力を同時に受信するように、ビデオ入力モジュール522を構成するのが好ましい。さらに、これらのビデオ入力により生成されたビデオ信号を、それぞれ、ライン584上にマルチプレクシングして、(その後、接続されたビデオ・ランプユニットにそれぞれ伝送するため)出力モジュール538に供給するのが好ましい。

【0272】このようにして、ビデオ・ランプユニットに、それぞれ、多くのビデオ信号を供給しプロジェクトする。ビデオ出力モジュール538に接続されている各ビデオ・ランプユニットと互換性のあるフォーマットに、これらのビデオ信号が変換されるように、ビデオ出力モジュール538が、それぞれ、構成されるのが好ましい。

【0273】デジタル・イメージファイルは、メインプロセッサ・カーネル502により、出力モジュールにそれぞれブロードキャストされるのが好ましい。しかし、イメージ生成ランプユニットをサポートする出力モジュール592のセットのモジュールのみが応答すべきである。

【0274】ランプユニット・プロセッサがイメージからイメージにビデオ「ディゾルブ(dissolve)」または「ワイプ(wipe)」するような機能を実行するため、ランプユニット・プロジェクション・ゲートでのイメージデータのある変化を指定する制御信号を、モジュラ・コントローラ・メインフレーム500からランプユニットに伝送することができる。このような「ディゾルブ」または「ワイプ」コマンドは、制御資源ネットワークの(図2に示す)両方向バス80について説明した方法で、グラフィックス・ワークステーションからか、あるいは、制御パネル546-552からオリジネートされる。

【0275】本実施例では、出力モジュール540は、

ACパワー調光器544を制御するため、DMX-512 シリアル・データリンク542をインプリメントする。DMX-512は、慣用的には、固定焦点ランプユニットに対して、最高512個のチャンネルをサポートし、512個未満のチャンネルを慣用のランプユニットに割り振ることができる。

【0276】本実施例では、演技で共通に利用されるランプの他に、複数の出力デバイスをサポートするように、出力モジュール533が構成されている。上述したように、他のステージ・アクション効果は、照明コンソールにより制御する必要がある。チェインホイスト・モータ66と、エアーキャノン68と、特殊効果プロジェクト70と、スモークマシンのオペレーションをディレクトするための制御信号を生成するため、例えば、既に図2について説明したように、制御信号コンバータ64と同様の方法で、出力モジュール533を構成することができる。

【0277】ステータス・レポート

メインプロセッサ・カーネル502に伝送するため、ランプユニットからのステータス・レポートを受信するように、ステータス・レポートを伝送することができるランプユニットをサポートする出力モジュール592のセットの各モジュールが構成されるのが好ましい。その後、制御デバイスに伝送するため、メインプロセッサ・カーネル502により、入力バス512を介して、入力モジュール590のセットの各モジュールに伝送することができる。

【0278】例えば、メインプロセッサ・カーネル502により受信されたステータス・レポートを、入力モジュール514により、制御パネル・ユニット546-552に中継するか、および/または、入力モジュール520によりパーソナル・コンピュータ560に中継することができる。これら制御デバイス、例えば、546-552、560のオペレーティングシステム・ソフトウェアにより、制御デバイスはこのようなレポートを受信し表示することができる。

【0279】その上、ステータス・レポートをモニタし分析するため、AI (artificial intelligence) プログラム、または、「エキスパート・システム」を、これら制御デバイス546-552のオペレーティング・ソフトウェアの一部としてインストールすることができる。このようにして、公知の症状と、可能性のあるフォールトのデータベースと、ステータス・レポートを比較して、制御デバイスはステータス・レポートを走査し、システムのフォールトを識別することができる。

【0280】よって、制御デバイス546-552が、例えば、大多数のランプユニットが個々のランプユニット・ヘッド・アセンブリのオーバ温度条件をレポートする場合、ランプユニットの周囲温度が正規の温度より低く高いと結論して通知することが可能になる。

【0281】メインプロセッサ・カーネル502からのステータス・レポートを受信すると、入力モジュール514は、トークンバス制御装置として具現化するのが好ましいが、ステータス・レポートを、制御パケットとともに、制御パネル・ユニット546-552のうちの1つに伝送する。そのメッセージを受信する第1制御パネル・ユニット、すなわち、ユニット548はそのレポートをオペレータに表示する。第2制御パネル・ユニット、すなわち、ユニット550に伝送するため、制御ユニット548のプロセッサは、制御パケットと、ステータス・レポートと、オペレータ制御アクションにより生成された任意の制御入力信号よりなるメッセージをコンパイルすることになる。

【0282】第2制御パネル・ユニット550はステータス・レポートを受信し、そのステータス・レポートを表示する。シリアル・トークンバス523に接続されている次のデバイス、すなわち、制御パネル・ユニット552に伝送するため、第2制御ユニット550のプロセッサは、制御パケットと、ステータス・レポートと、第1制御パネル・ユニット548により付加された制御入力信号と、第2制御パネル・ユニット550でオペレータ制御アクションにより生成された制御入力信号よりなるメッセージをコンパイルする。

【0283】結局、メッセージはトークンバス制御装置514に返される。その後、メインプロセッサ・カーネル502に伝送するため、トークンバス制御装置514はステータス・レポートを破棄し、制御入力信号をコンパイルすることになる。コンピュータ制御される照明システムを分散処理によりオペレートするため、ある状態が異なる制御を必要とするという経験(experience)を示した。リハーサル・モードでは、例えば、ランプユニットを選択し、マルチプル・パラメタを操作するための拡張制御が、システムをプログラムして所望の照明効果を奏するために必要である。一方、演技モードでは、キューデータをリコールし、ランプユニットのグループを制御する拡張制御が、本システムをオペレートして、所望の照明効果をプレイバックおよび/または組合わせるために必要である。

【0284】シンクロナイズド演技モードでは、モジュラ・コントローラ・メインフレーム500が入力モジュール590のセットのモジュールの1つを介してタイミング信号を受信し、キューを、タイムコード・プログラムに従ってリコールし、所望の照明効果がショーの他のイベントと協動して再生されるとき、制御は必要でない。

【0285】さらに別の演技モードでは、モジュラ・コントローラ・メインフレーム500に接続された制御パネル・ユニット546-552に、簡単なディスプレイ手段と、マニュアル・オーバライド(override)または緊急つまみを提供し、タイミング信号が失われた場合であ

っても、所望の照明効果を再生することができることは、適正なことである。

【0286】そのため、本発明は、コンピュータ制御照明システムのための制御システムに、照明システムに用いられる具体的なアプリケーションに依存して異なるコンソールを再構成および／または置換することができる分散処理を提供する。

【0287】可能な全ての制御を常に利用可能にする1つのコンソールを提供することは、最早、必要ではない。詳しく言うと、本発明の照明制御システムと、他の電氣的な制御システムとの電気制御インタフェースを提供する種々の要件を含み、リハーサル・プログラミングと演技ブレイバックという異なる要件に適合させるように、制御システムを再構成することができる。

【0288】以上、本発明の幾つかの実施例を、発明の実施の形態において図面を参照して説明したが、本発明は開示した実施例に限定されるものではなく、本発明の範囲を逸脱しない限り、種々に再構成し、修正し、置換することができることは、当然のことである。

【0289】別の実施例では、別種の制御システム要素のより効率的な利用を容易にするため、開示された照明システムは分散制御システムにより制御される。上述したモジュラ制御資源ネットワークは、同時にオペレートして複数の自動化および慣用ランプユニットを制御するマルチプル・オンライン制御パネルのために設けられ、モジュラ・コントローラ・メインフレームは、依然、制御装置とランプユニットの間のデータリンク通信の単一の複雑なノードである。

【0290】モジュラ・コントローラ・メインフレームが複雑であるので、通信は障害を起こし易く、通信の中央ノードとしての役割を担っているため、その通信障害は、危険をはらんでいる。制御システムの中央コンポーネント、例えば、上述した制御資源ネットワークでの通信障害の危険性を除くため、簡略化された高速データリンク・ネットワークにより相互結合された分散制御システム要素により特徴付けられた分散制御システムが利用される。

【0291】図28を説明する。分散制御システムは、既に記述したタイプの1つ以上の制御コンソール24A、24B、24Cと、1つ以上の汎用コンピュータ560A、560B、560C、560Dと、典型的には1つ以上のインタフェース・モジュール602、610A、610B、610Cであるが、パワーおよび信号配分ラック400(図17)と必ずしも関係付けをしていないインタフェース・モジュールとよりなる。上述した制御システム要素は全て高速データ配分ネットワーク600、例えば、Ethernetを介して通信が行われる。

【0292】種々の制御システム要素はネットワーク上のピア(peer)として存在することができる。すなわち、どの要素もネットワーク上でデータ通信を制御していな

いが、全ての要素はネットワークに同等にアクセスすることができる。2つ以上の制御システム要素がデータを同時に伝送したために生じたコンフリクト(conflict)を解決するか除去するために、幾つかの良く理解されたスキームが利用される。あるいはまた、図27の説明で記載したようなトークンバスを利用することができる。

【0293】制御コンソールおよびパーソナル・コンピュータをローカルで接続し、両者を高速制御資源ネットワーク600に接続する他の構成を利用することができる。

【0294】1つのシステム要素はシステムデータ・レポジトリとして一意のステータスを有する。この1つのシステム要素は、照明システム、例えば、上記特許出願第07/641,031号(米国特許第5,307,295号、発明の名称:Creating and Controlling Lighting Designs)に記載されたような照明システムの3次元モデルをストアする汎用コンピュータ560でも良い。

【0295】図29は制御コンソールの典型的なアーキテクチャを示す。制御コンソール24は照明システムのランプユニットを直接制御するように適正化されているが、可変照明パラメタをマニュアル制御することと、後程リコールされるパラメタ・データをストアすることと、ストアされたパラメタをリコールしこのパラメタに従うようにディレクトすることに限定されない。コンソールは主に制御コマンドをランプユニットに送信するためにあり、制限された表示能力を含むのが典型的である。制御パネル664は、種々のスイッチと、ノブと、フェーダと、人間のオペレータが本システムのオペレータをディレクトするのに用いられるディスプレイを含む。アドレス、データ、制御信号バス654に接続されたマイクロプロセッサ650は、ROM(Read-Only Memory)652にストアされたプログラムを実行し、本システムのオペレーションに関連するデータを、RAM(Random Access Memory)652にストアする。RAMおよびROMはバス654に接続されている。マイクロプロセッサ650は、バス654に接続されている制御インタフェースを介して制御パネル664を解釈する。タッチ検知ディスプレイ662はバス654に接続されているディスプレイ・インタフェース回路660に接続されている。マイクロプロセッサ650はタッチ検知ディスプレイ662からのタッチ入力をインタフェース660を介して読み取り、ディスプレイ・データを同一インタフェースを介してディスプレイに書き込む。高速データリンク・トランシーバ658はバス654に接続され、コンソールと、高速制御資源ネットワーク600とのインタフェースとしてのサーバに接続される。

【0296】図30は汎用コンピュータの典型的なアーキテクチャを示す。マイクロプロセッサ620はROM622にストアされているプログラムを実行し、ハードディスク・ドライブにストアされているプログラムをR

AM6 2 3にロードし、RAM6 2 3からのプログラムを実行する。アドレス、データ、制御信号バス6 5 4は、マイクロプロセッサと、RAMと、ROMと、ハードディスク・ドライブと、次に説明する他のコンポーネントと相互結合されている。リムーバブル・ディスクドライブ6 2 6、例えば、光磁気ハードディスク・カートリッジは、ショーファイルのバックアップ・コピーをトランスポート可能な形態でストアする。ディスプレイ・インタフェース6 3 0はバス6 2 4に接続されており、グラフィックス・ディスプレイ・デバイス6 3 2のオペレーションをサポートする。高速データリンク・トランシーバ6 2 8はバス6 2 4に接続されており、コンピュータと、高速制御資源ネットワーク6 0 0とのインタフェースとしてサージする。入力デバイスは、インタフェース6 3 6と、ポインティング・デバイス6 4 4（例えば、マウスか、ライトペンか、トラックボールか、インタフェース6 3 6を介して接続された同様のデバイス）と、マイクロホン付き音声認識モジュール6 4 0（インタフェース6 3 4を介して接続されている）を介して接続された標準キーボード6 4 2を含むことができる。

【0 2 9 7】汎用コンピュータ5 6 0は、データを、パラメタデータのテーブル状のビューを含むとともに、演技環境または他の照明環境のグラフィカル・ビューを含めて、ストアし、表示する。あるシステムの1つの汎用コンピュータは、ランプユニットと、照明される物体を含めて、演技環境または他の照明環境の3次元モデルを含む「ショー・ファイル」をストアするシステムデータ・レポジトリである。このショーファイルにも、これらランプユニットに対するパラメタ・データが含まれる。ショーファイルは、ランプユニットにストアされているパラメタ・データセット（キューデータ）に対するバックアップ・データとしてサージする。このショーファイルは後程説明する他の目的もサージする。次に説明するが、ショーファイルのコピーは本システムの他の汎用コンピュータに存在させることができる。システムデータ・レポジトリのショーファイルが更新されたときにこれらのコピーを更新する手段を採用することができる。本システムの任意の汎用コンピュータは、ランプユニット・ステータス情報をディスプレイすることができ、システムデータ・レポジトリの任意のショーファイルをビューし、作成し、エディットすることができる。汎用コンピュータは、当該コンピュータのディスプレイの「ウインドウ」の仮想制御コンソールを作成するプログラムもランさせることができる。そのウインドウを利用しているコンピュータから、ポインティング・デバイス6 4 4および/またはキーボード6 4 2と協働して、人間のオペレータはランプユニットを制御することができる。

【0 2 9 8】図3 1および図3 2は負荷インタフェース・モジュールの典型的なアーキテクチャを示す。負荷インタフェース・モジュール6 0 2、6 1 0は、主に、負

荷インタフェース・モジュール6 0 2、6 1 0に接続されている全てのランプユニットに、コマンドをファンネル(funnel)し、これら全てのランプユニットのステータスをレポートする。制御コンソールおよび汎用コンピュータによりディスプレイするため、負荷インタフェース・モジュールは周期的にステータス・データを高速制御資源ネットワークを介して伝送する。負荷インタフェース・モジュールは、マルチプル・デバイス制御ネットワークへのネットワーク・ゲートウェーでもあり、高速制御資源ネットワークを介して、ある1つのフォーマットで、データを伝送し受信し、一方、個別のデバイス制御ネットワークを介して、ある別のフォーマットで、データを伝送し受信する。

【0 2 9 9】負荷インタフェース・モジュール6 0 2、6 1 0は、それぞれ、独立のデータネットワークとして構成されており、高速分散制御資源ネットワーク6 0 0と、別種の通信プロトコルと機能とデータフォーマットを有する1つ以上のタイプのランプユニットとのインタフェースとしてサージすることができる。負荷インタフェース・モジュールは、それぞれ、必要な場合には、制御コンソールまたは汎用コンピュータから受信されたジェネリック・コンソール・コマンドを、具体的なタイプのランプユニットか、あるいは、当該ランプユニットに接続されている出力デバイスと通信するのに必要な具体的なコマンドまたはパラメタに変換する。

【0 3 0 0】負荷インタフェース・モジュール6 0 2（図3 1）は、マイクロプロセッサ6 7 0と、このマイクロプロセッサをオペレートする実行可能なプログラムを含むROM6 7 2と、そのインタフェースのオペレーションに関連するデータをストアするためのRAM6 7 3と、高速制御資源ネットワーク6 0 0に接続するためのデータリンク・トランシーバ6 7 8と、インタフェース回路6 7 5に接続されているフロントパネル調整つまみおよびインディケータ6 7 6と、独立のデバイス制御ネットワークをサポートするランプユニット・インタフェース6 8 0とを含む。アドレス、データ、制御信号バス6 7 4は、負荷インタフェース・モジュール6 0 2の主機能ブロックと相互結合されている。

【0 3 0 1】負荷インタフェース・モジュール6 1 0（図3 2）は図2 2に示すスマート・リピータ・ユニットに対応しており、制御コンソールと高速データリンク6 0 0を介して通信を行うように修正されている。図2 3に示すManchesterエンコーダ4 7 0およびManchesterデコーダ4 7 2は、負荷インタフェース・モジュール6 1 0において、高速データリンク6 0 0に接続するため、高速データリンク・トランシーバ6 1 2と置換される。あるいはまた、両アップリンク・インタフェースを1つのカード・アセンブリと、エンコーダ4 7 0と、デコーダ4 7 2に含むことができ、同様に、トランシーバ6 1 2に含むことができる。よって、1つの標準回路カ

ード・アセンブリはスマート・リピータ56（図24および図25）としてサージすることができ、負荷インタフェース・モジュール610（図28）としてサージすることができる。

【0302】高速制御資源データ・ネットワーク600は工業標準であって、市販のLAN（local-area-network）フォーマットであり、当該ネットワーク上のデバイスに接続するためのポートを有するハブ・ユニット604、605、606として構成されている。回路カード・アセンブリのプラグとして構成されたトランシーバ回路628、658、678、612は、各コンソールと、汎用コンピュータと、負荷インタフェース・モジュールにインストールされる。あるいはまた、制御コンソールの機能をインプリメントするか、あるいは、インタフェース・モジュールにロードするように設計された他の回路カード・アセンブリに、トランシーバ回路を構築することができるが、改善されたネットワーク・フォーマットが利用可能になるか、あるいは利用可能になった場合、カード構成のプラグにより、トランシーバ回路を容易に除去することができ、改良されたトランシーバ回路と置換することができる。

【0303】例えば、ステージまたは他の演技領域か、あるいは、照明制御ブースに隣接した観客席に設けた制御コンソールおよび汎用コンピュータに、1つのハブユニット604が並置されているのが好ましい。制御コンソール24A、24Bと、汎用コンピュータ560A、560Bは、ハブユニット604に接続されている。本発明に係る負荷インタフェースを含むパワーおよび信号配分ラックに並置された第2ハブユニット605に接続されている。負荷インタフェース610A、610B、610Cは第2ハブユニット605に接続されており、コンソールおよび汎用コンピュータにより生成された制御信号を、負荷インタフェース610A、610B、610Cに接続されているランプユニットに供給されている。第2ハブユニットに接続された第3ハブユニット606は、多大なランプユニットををサポートするために設けられている。ハブユニット間の接続は、高バンド幅のリンク、例えば、光ファイバ・ケーブル通信フォーマットでも良いし、ハブユニットと、コンソールと、コンピュータと、負荷インタフェース・モジュールとの接続はワイヤ・リンクでも良い。

【0304】追加の制御システム要素、例えば、機能限定テクニシャン用コンソール24Cか、あるいは、リモート・フォーカシング・デバイス608を、舞台または演技領域と並置された第2、第3、および他のハブユニットに接続することができる。他のハブユニットを、他のコンソールと並置されていない追加のコンソールに接続することができる。

【0305】オペレーションの方法

マルチプル制御コンソールを、同時に、オンラインで、

コンピュータ制御される本発明に係る照明システムで、オペレートする方法は、並行コマンドを、同一のランプユニットまたはランプのグループに発行する2つ以上のコンソールのコンフリクトを解決しなければならない。オペレーションのデフォルト・モードでは、あるシステムのランプユニットを、それぞれ、当該システムの任意のコンソールにより制御することができる。

【0306】ランプユニットの異なるグループを同時にプログラミングすることができるように、マルチプル制御コンソールを同時にオンラインにすることができる。当該システムのランプユニットを、マルチプル・コンソールにより同時にマニュアル制御するために選択することができる。ユーザが選択可能なオプションにより、マニュアル制御のために選択されたランプユニットを、別のコンソール80、すなわち、第1コンソールで選択解除され、マニュアル制御のため、第2コンソールにより選択された別のコンソール80により捉えることができるか否かを判定するか、あるいは、第2コンソールによりマニュアル制御するため、ランプユニットを選択することができる前に、そのランプユニットが第1コンソールの人間のオペレータにより選択解除されるまで、第2コンソールが待機しなければならないか否かを判定する。キューをストアしリコールするため、当該システムのランプユニットを、当該システムの任意のコンソールまたはコンピュータにより、ディレクトすることができる。あるいはまた、制御のために、あるランプユニットが排他的にあるコンソールに割り当てられるように、キュー・ストアおよびリコール・オペレーションを制限することができる。

【0307】オペレーションの第2の「ランプユニット・ロックド(lamp-unit-locked)」モードでは、任意のコンソールは特定のランプユニットか、あるいは、当該コンソールにのみ応答するランプユニットのグループをレンジリングするコマンドを発行することができる。オペレーションのこの第2のモードでは、他のコンソールはランプユニットまたはランプユニットのグループに影響を与えることはできない。ランプユニット・ロックド・モードを用いて、照明システムを、あるコンソールのみで制御されているランプユニットのグループに区分することができる。各グループ内では、ランプユニットはキュー・ストアに応答し、ランプユニットが「ロック」されている1つのコンソールのみからコマンドをリコールする。

【0308】オペレーションの第3の「ランプ機能ロックド(lamp-function-locked)」モードでは、任意のコンソールは、ランプユニットまたはランプユニットのグループのある機能、例えば、パンおよびチルト制御のような機能を、当該コンソールのみに応答して、レンジリングするコマンドを発行することができる。一方、ランプユニットの他の機能、例えば、ビーム色およびビーム光

度のような機能を他の任意のコンソールにより制御することができる。

【0309】オペレーションの第4の「ランプユニット・リミテッド(lamp-unit-limited)モードでは、制御コンソールを、ランプユニットのあるグループのみを制御するように制御することができ、同様に、ランプユニットを他のコンソールにより制御することができる。ユーザ選択可能オプションにより、機能限定コンソールからマニュアル制御により現在選択されたランプユニットを、別のコンソールにより捉えることができるか否かが判定されるか、あるいは、ランプユニットが機能限定コンソールで選択解除されるまで、他のコンソールは待機しなければならないか否かが判定される。機能限定コンソールによりリコールされたキューでアクティブになるランプユニットが、他のコンソールによりリコールされたキューに応答することになるか否かが、ユーザ選択可能オプションにより判定されるか、あるいは、機能限定コンソールによりリコールされたキューでランプユニットが現在アクティブである場合、そのランプユニットは他のコンソールからのキュー・リコール・コマンドを無視するか否かがユーザ選択可能オプションにより判定される。

【0310】オペレーションの第5の「ランプ機能限定(lamp-function-limited)」モードでは、制御コンソールを、ランプユニットのある機能のみの制御に限定することができるが、これらの機能を他のコンソールにより制御することができる。マニュアル制御のために選択されているか、あるいは、機能限定コンソールによりリコールされるアクティブ・キューにより制御されているランプ機能を、別のコンソールにより捉えることができるか否かがユーザ選択可能なオプションにより判定されるか、あるいは、これらの機能が機能限定コンソールにより現在制御されている場合に、他のコンソールからのコマンドをランプユニットが無視するか否かがユーザ選択可能なオプションにより判定される。

【0311】ボードレベル・キューまたは同様のマクロ機能により、特定の制御構成をセットアップすることができる。単一の制御コンソールに対してオペレート可能なボードレベル・キューか、あるいは、汎用コンピュータでランする仮想制御コンソールは、種々のフロントパネル調整つまみの公知のステートまたは所望のステートのセットを表すデータセットであり、機能限定コンソールのフロントパネル上のインディケータである。人間のオペレータはコンソール・フロントパネル機構に所望の公知のステートを迅速に設定するため、ボードレベル・キューをリコールすることができるが、コンソール・フロントパネル機構に所望のステートを設定するため、数多くのフロントパネル調整つまみを個々に操作することはできない。ボードレベル・キュー・リコールは、人間のオペレータがキーパッドを操作しついで“recall”ボタ

ンを押下して入力する2つまたは3つのキャラクタ・ボード・キュー識別子の形態をとることができる。キュー数値またはチェース数値を幾つかのサブマスタにロードする場合に、必要と思われる16回ないし20回か、あるいはそれ以上、ボタンを押下していたのに対して、このような識別子の形態をとると、ボードレベル・セットアップをするのにボタンを3回または4回押下するだけでよい。

【0312】オンラインで、マルチプル制御コンソールで同時にオペレートするボードレベル・キューは、異なるランプユニットまたはランプユニットのグループに対して、ランプユニット・ロッキングと、ランプ機能ロッキングと、ランプユニット・リミティングと、ランプ機能リミティングという種々のモードを含むことができる。

【0313】マルチプル・コンソールとショーファイルとの対話

本発明に係る分散制御コンソールでは、1つ以上の汎用コンピュータと、オンラインで、同時に、マルチプル制御コンソールをオペレートするための方法は、システムデータ・レポジトリに維持されているショーファイル・データベースに影響を与えるコマンドを発行する2つ以上のコンソールの間でのコンフリクトを解決しなければならない。第2以上の汎用コンピュータを含むシステムでは、1つ以上のショーファイル・データ・エディティング・プログラムは、当該コンピュータで同時にランさせることができる。その例としては、マルチプル制御コンソールと、ショーファイル・データ・エディタ・プログラムをランさせているマルチプル・コンピュータとが、同時にオンラインになることができる機会が存在する。

【0314】オペレーションのデフォルト・モードでは、現在オープンされているショーファイルであって、制御資源ネットワークに接続されているショーファイルを、任意の制御コンソールによりエディットすることができるか、あるいは、当該システムのショーファイル・データ・エディタ(ショー・エディタ)によりエディットすることができる。

【0315】オペレーションの第2の「ショーファイル・ロックド(show-file-locked)」モードでは、任意のコンソールまたはショー・エディタは、当該コンソールまたはショーエディタのみによるエディット可能なショーファイルをレンダリングするコマンドを発行することができる。この第2のモードでは、1つのコンソールのみがキューをストアすることができるか、あるいは、1つのショー・エディタのみがデータをショーファイルに追加するか、ショーファイルからデータを削除するか、ショーファイルのデータを修正することができる。

【0316】オペレーションの第3の「レコード・ロックド(record-locked)」モードでは、任意のコンソールまたはショーファイルは、あるレコード、またはレコー

ドのグループ、例えば、あるキューの全てのランプユニット・レコードか、あるランプユニットのランプユニット・レコードか、あるいは、他の幾つかのグルーピングであって、当該コンソールまたはショーエディタのみによりエディット可能なものをレンダリングするコマンドを発行する。オペレーションのこの第3のモードでは、当該システムの任意のコンソールまたはショーエディタは、特定のコンソールまたはショーファイルに対してロックされているレコードを除いて、当該ショーファイルの全てのデータに対してオペレートすることができる。

【0317】ショーファイルの更新

キューはランプユニットにストアされているが、連続してオペレートしているバックグラウンド・ルーチンは、ランプユニットにストアされた分散データベースの変更に従って、ショーファイルを更新することができる。

【0318】ショーファイルは、汎用コンピュータでランしているショーエディタにより直接エディットすることができる。ショーファイルのレコードがエディットされるように、バックグラウンド・ルーチンは、ランプユニットにストアされた分散データベースを、ショーファイルの変更に応じて更新する。

【0319】異なるタイプの制御コンソール

個々の制御パネルに含まれる異なる制御デバイスを有する複数の制御コンソールに対して、それぞれ、オペレーションの適正なモードと、上述したようなユーザ選択可能なオプションをエレクトして、異なる機能のために適正化された種々の制御コンソールをインプリメントすることができる。テクニシャン・コンソール24Cの機能を、テクニシャン・オペレータがランプを点灯し消灯することができ、光度制御およびパン制御の変動を小さくすることができるように制御することができるが、色を制御したり、チルトまたは他の機能を制御したり、キューをストアしたりリコールしたりすることができないように制限することができる。リモート・フォーカシング・デバイス608の機能を、照明ディレクタが基準キューでアクティブ・ランプを逐次化することができるか、あるいは、キューをディレクトし、フォーカス（パン、チルト、ビームサイズ、エッジ/オブティカル・フォーカス）を調整することができるように制限することができる。

【0320】マスタコンソールはデフォルト・モードでオペレートし、ランプユニット・ロックド・モードで2次コンソールに区分されていない、システムの全ランプユニットを制御することができる。照明システム全体のサブセット、例えば、ワーク・ライトか、ハウス・ライトか、あるいは、メインステージまたは演技場を照明するのに使用されない他のランプユニットを制御するため、2次コンソールはランプユニット・ロックド・モードでオペレートすることができ、一方、マスタ・コンソールはメインステージの照明を制御する。周辺ランプ

ユニットを2次コンソールに区分すると、メインコンソールのオペレータは、マニュアル制御のために全てのランプを選択することができ、メインステージ照明用に指定されたランプユニットのみを制御することができ、周辺ランプユニットの制御は免れる。

【0321】さらに、ハブユニット606のようなハブユニットは、コンソールと、負荷インタフェース・モジュールと、当該ハブに接続されたコンピュータとの間のデータリンク・トラフィックを、他のハブに接続している制御コンソール間のデータリンク・トラフィックから分離することができる。このようにして、テクニシャン・コンソール24Cおよび汎用コンピュータ560Cは、米国特許5,282,121号に開示されているような液晶プロジェクション・ゲートアレイを備えたランプユニットによるプロジェクトのための適正なランプユニット・イメージ・ファイルを設計し、エディットし、その適正なランプユニット・イメージ・ファイルにダウンロードすることができる。しかし、過度の不適正なデータリンク・トラフィックを高速制御コンソール・ネットワーク600の他のセグメントにはもたらさない。

【0322】設計者のリモート・コンソールはディスプレイ・オンリ・デバイスであり、このデバイスから、デバイスのユーザはデータの表示をディレクトすることができる。設計者のリモート・コンソールはシステム・データ・レポジトリと、高速データリンク・ネットワークを介して通信を行なうことができ、当該システムのランプユニットの現在のパラメータを表示することができ、ショーファイルの任意のキューに対するキューデータを表示することができ、マスタまたは1次制御コンソールで用いられるディスプレイ・ウインドウを表示することができる。

【0323】ディスプレイのコンセプト

本発明に係る分散制御システムは、照明システムのグラフィカル表現を、種々のフォーマットで表示する能力を持っている。照明のシステムの3次元モデルであって、システムデータ・レポジトリに保持されているショーファイル・データベースに含まれているモデルに基づき、グラフィック・ディスプレイ・フォーマットは、システムの現在の状態を表す情報か、あるいは、当該モデルでキューがリコールされたときは、当該システムの可能なステートを表す情報をオペレータに運ぶ。

【0324】フォト・リアリスティック・ディスプレイ
1つのグラフィック・ディスプレイ・フォーマットは、ステージと、セット・ピースと、演技者のフォト・リアリスティック・ビューを提供し、それらがランプユニットによる現在の照明でどのように見えるかをフォト・リアリスティック・ビューで提供する。照明デザイナーの通信またはセールス・ツールとしてシミュレートするために、このビューを用いることができる。リアリスティック・フィードバックを、オフライン・ショー・プログラ

ムに提供し、しかも、ステージを遮る制御ロケーションにも提供するためにも、このビューを用いることができる。フォト・リアリスティック・ビューでは、円錐状または円柱状のビームをプロジェクトした場合に、空中にスモークが存在しているように思える程度に、このようなビームをプロジェクトしてランプユニットを表現する。1つのランプユニットによりプロジェクトされた光のプールか、あるいは、遮光板(gobo)のパターンは、そのランプユニットにより照明されたサーフェスにマッピングされている。種々のサーフェス・マテリアルが、照明へのはね返りと、そのサーフェス・マテリアルの反射プロパティを考慮して、このモデルで採用されている。モデル化されたスクリーンで実際のビデオ・シーケンスをプレイすることができるオペレータは、ビデオ・スクリーンをこのモデルに含めることができる。キューがこのモデルでリコールされるとき、フォト・リアリスティック・ビューは空中にビームが円滑にアニメートされる様子を示し、ビームが動くにつれてプロジェクションが円滑にアニメートされる様子を示す。どの角度からでもそのモデルをビューすることができるよう、オペレータの観点を迅速かつ容易に移動させることができる。オペレータは、1つのボタンを押下するだけで、プリセットされた観点に移動させることができる。

【0325】ワーキング・ビュー・ディスプレイ

もう1つのグラフィック・ディスプレイ・フォーマットは、リアリスティックの程度が劣る3次元グラフィックスと、2次元グラフィックスと、数値データのミックスを備えたモデルのワーキング・ビューを提供する。3次元グラフィックスはビジュアル・コンテキストを他のデータに提供する。2次元グラフィックスは、ビジュアルと品質に関するランプユニット情報を提供する。数値情報は品質に関するランプユニット情報を提供する。ステージと、セット・ピースと、演技者には、よりブロック・ライク(block-like)なやり方で、現実味の点で劣るが、提供される。組込み型ビデオ・スクリーンは含まれないし、サーフェス・モデリングも含まれないし、ランプユニットのビームのモデリングも、プロジェクションも含まれない。アクティブ・ランプユニットによりプロジェクトされたライト・ビームは、カラー・バーにより表現される。カラーバーに対するグラフィカルな影響により、パターン・プロジェクション(遮光板)と、ハードまたはソフトのエッジ効果(オブチカル・フォーカス)と、「マーク付けした」ランプユニット(全ての可変パラメタと一致するが、所定のキューで、光ビームが実際にはプロジェクトされないユニット)が表現される。、ランプユニットが指す目標を示すため、色の付いたドットか、あるいは、他のグラフィカル・デバイスにより、ビームの終点(terminus)を表現することができる。マニュアル制御するため、マウスまたは他のポインティング・デバイスでビーム・バーをクリックして、対

応するランプユニットを選択する。ポインティング・デバイスでビーム終点をドラッグして、ランプユニットをフォーカシングし直すことができる。光ビームの狙いを簡単にするため、ビーム終点を折り曲げて目標サーフェスまたは演技者の頭に向けるか、あるいは、他のデフォルト・ポイントに向ける。一度、ビームのジオメトリック・パターンを構成すると、光ビームをグループ化することができ、その後、そのグループの任意の光ビームをドラッグしただけで、グループとして移動させることができる。マニュアル制御するため、光ビームのグループをクリックして、ランプユニットの対応するグループが選択される。

【0326】キューがリコールされると、ワーキング・ビューは、ビームバーが空中で移動するにつれて、そのビームバーの大雑把なアニメーション様の動きを示す。どの角度からでもモデルをビューすることができるよう、オペレータの観点を迅速かつ容易に移動させることができる。オペレータは、1つのボタンを押下するだけで、プリセットされた観点に移動させることができる。

【0327】オペレータは、ステージの前縁から少し離れた位置から、ステージとオーバヘッド照明トラスの間までの魚眼(fish-eye)ビューを得ることができ、下から見たオーバヘッド照明トラスのビューと、上から見たステージのビューを得ることができる。オペレータの観点の動きは、魚眼ワーキングビューのうちのステージの前縁を中心とした円内に限定される。ワーキングビューは、選択された単一のパラメタの数値ディスプレイを、各ランプユニットの本体上に重ねることができる。オペレータはランプユニット・ビームバーのグループとをON、OFFすることができる。ランプユニットを「ページ」で表示することができ、(典型的には100)連続するランプユニット・アドレスのグループで表示することができる。マニュアル制御のために、対応するランプユニットを選択するように、オペレータがビームをクリックしたとき、選択されたランプユニットと同一のビーム・カラーを有する全てのアクティブ・ランプユニットも、マニュアル制御のために選択されるように、「マジック・ワンド」選択ツールが提供される。このマジック・ワンド・ツールを用いて、オペレータがターゲットをクリックすると、当該ターゲットにフォーカシングされた全てのアクティブ・ランプユニットがマニュアル制御のために選択される。マニュアル制御のために選択されたランプユニットも数値ディスプレイのために選択され、オペレータは、ランプユニットのうちの幾つかのランプユニットのみの数値データを、ページで、ビューすることができる。

【0328】チャンネル・ユーセジ・ディスプレイ(Channel-usage Display)

照明システムと、ステージと、セット・ピースと、演技者の3次元モデルを維持している間、そのモデルを破壊

するビューを提供するチャンネル・ユーセジ・ディスプレイを、別のグラフィック・ディスプレイ・フォーマットは提供する。チャンネル・ユーセジ・ディスプレイは、ステージのリアリスティック・ビューではなく、データ・ビジュアライゼーション・ツールとして意図されており、チャンネル・ユーセジ・ディスプレイを操作して、ステージの重要なサーフェスと、セット・ピースと、バックドロップ(backdrop)を、1つのビューで示することができるが、セットの他の部分で隠されているセットの部分を見るため、オペレータの観点を移動させる必要はない。オペレータの観点はチャンネル・ユーセジ・ディスプレイでは固定されているが、そのモデルの部分をそのビュー内で移動させ回転させて、それらの重要なサーフェスを固定観点から見えるようにすることができる。

【0329】例えば、セット・ピースをステージの前から見ることができ、ステージの面が見えるようにステージを下方に傾けることができ、バックドロップをセット・ピースの背後から持ち上げて、バックドロップの下部が見えるようにすることができる。ランプユニット自体は見えず、空中のライトビームも見えない。セット・ピースまたは他のサーフェスにプロジェクトされたライトビームの輪郭をマーク付けするラインが描かれる。そのサーフェスにプロジェクトされたビームにより形成された光のプールの輪郭を描くため、幾つかのグレイイング効果(graying effect)を用いることができる。ランプユニットまたは制御チャンネル番号を、ビームがプロジェクトされた領域の中央部にディスプレイすることができるか、あるいは、ランプユニットまたは制御チャンネル番号を、複数ビームが単一のポイントに収束した場合には、ターンオフすることができる。ビームがプロジェクトされる対応する領域か、アドレスまたはチャンネル番号か、あるいは、ターゲットを、マウスまたは他のポインティング・デバイスを用いてクリックして、マニュアル制御のため、ランプユニットを選択することができる。

【0330】コンソール機構

本発明に係る分散制御システムで利用される制御コンソールは、次の1)ないし7)のような機能を含む種々の機能に対して適正化された制御デバイスを含むことができる。1) マニュアル制御のために、ランプユニットまたはランプユニットのグループを選択または選択解除する機能と、2) マニュアル制御のために選択されたランプユニットの可変パラメタを調整する機能と、3) その後リコールするため、本システムのランプユニットの現在のステートを記述する可変パラメタをストアする機能と、4) 可変パラメタ・ランプユニットをリコールして、ストアされている可変パラメタ・データに一致させる機能と、5) スタアされている可変パラメタ・データをランプユニットからシステムデータ・レポジトリにアップロードする機能と、6) スタアされている可変パラメタデータをシステムデータ・レポジトリからランプユ

ニットにダウンロードする機能と、7) 本システムのランプユニットの現在のステートに関連するステータス情報を要求しディスプレイする機能。

【0331】幾つかの制御コンソール、例えば、Vari Lite, Inc of Dallas, Texas 製のArtisan(登録商標)およびmini-Artisan(登録商標)制御コンソールは、変数パラメタの単一のセットであってストアされている単一のセット(ダイレクト・キュー)と、可変パラメタデータのセットであってストアされたセットの自動実行可能なシーケンス(チェース)とをリコールする個別のフロントパネル機構を用いており、一方、本発明に係る制御コンソールはダイレクト・キューか、チェース・シーケンスか、あるいは他のプレイバック機能のいずれかをリコールすることができる柔軟なフロントパネル機構を利用する。

【0332】ダイレクト・キューをリコールするか、チェース(ダイレクト・キューのシーケンス)をリコールし実行するか、2つのダイレクト・キュー間でクロス・フェードを制御するか、あるいは、マトリクス・フェーダのグループに分配されたランプユニットのグループの光度を制御するかのいずれかのために、「サブマスタ」と呼ばれるフロントパネル機構が適合されるのが普通である。サブマスタに割り当てられたランプユニットの全体的な光度レベルを制御するために用いられる可変コントローラすなわち「フェーダ」と、当該サブマスタにダイレクト・キュー番号をロードするための「go」ボタンまたは当該サブマスタにチェース・シーケンス識別子をロードするための「load」ボタンと、当該サブマスタにロードされるダイレクト・キュー番号をインクリメントするかデクリメントするためのボタン対またはチェース・シーケンスでキューをランし停止するかシングル・ステッピングするためのボタン対を、サブマスタに提供するのが普通である。サブマスタにリコールされたキューまたはチェースをアクティベートするかデアクティベートするために、サブマスタ「select」ボタンが提供され、実行前に、当該キューまたはチェースをロードすることができ、ついで、適正な時点で、1つのボタンを押下して実行することができる。

【0333】本発明に係るコンソールは、ソフト・サブマスタ、例えば、図33に示すようなサブマスタを利用する。このソフト・サブマスタは、ダイレクト・キュー・サブマスタか、チェース・シーケンス・サブマスタか、他のサブマスタのうちのいずれかに、ダイレクト・キュー番号か、チェース・シーケンス識別子か、他の識別子をどのように当該サブマスタにロードするかによって、応答することができる。オペレータがキュー番号を数値キーパッドからエンタし、その数値キーパッドに関係付けをした「direct」ボタンを押下し、ついで、ソフト・サブマスタに関係付けをした「go」ボタンを押下した場合は、当該サブマスタは自動的にダイレクト・キュー

ー・サブマスタになり、当該サブマスタに関係付けをした他のボタンは、これらの他のボタンが適正な機能、例えば、当該サブマスタにリコールされたキュー番号をインクリメントするかデクリメントするような機能をパフォームするステートをとる。オペレータがチェース・シーケンス識別子を数値キーパッドからエンタし、当該数値キーパッドに関係付けをした「chase」ボタンを押下し、ついで、ソフト・サブマスタに関係付けをした「go」ボタンを押下した場合は、当該サブマスタは自動的にチェース・シーケンス・サブマスタになり、当該サブマスタに関係付けをした他のボタンは、これら他のボタンが適正な機能、例えば、チェース・シーケンスをランし停止するか、あるいは、キューのシーケンスをシングル・ステップするような機能をパフォームするステートをとる。

【0334】1つの数値を数値キーパッドからエンタし、特定のプレーバック機能を識別するボタンを押下して、他のプレイバック機能、例えば、2つのダイレクト・キューと、ランプユニットのグループの光度の単一のダイレクト・キューでの個々の制御（マトリクス制御）との間でクロス・フェードするような機能をパフォームするように、ソフト・サブマスタのグループを構成することができる。例えば、オペレータはキュー番号をキーパッドからエンタし、このキーパッドに関係付けをした「X-fade」ボタンを押下し、ソフト・サブマスタで「go」を押下し、ついで、別のキュー番号をエンタすることができ、「X-fade」ボタンを押下し、別のソフト・サブマスタで「go」を押下して、サブマスタ対を、2つのダイレクト・キューの間でフェードするクロス・フェード制御装置として構成することができる。

【0335】マトリクス・フェーダおよびマトリクス・サブマスタを指示するのに、同様の技術を採用することができる。マトリクス・サブマスタはランプユニットの複数のグループの全体的な光度を制御し、他のソフト・サブマスタと関係付けをしたマトリクス・フェーダの複数のグループのうちの1つのグループにより制御されているランプユニットのグループの全体的な光度を制御する。

【0336】ソフト・サブマスタに関係付けをしたボタンに付けたラベルを、必要とする機能により変更するため、本発明に係るコンソールは、プログラマブル・ディスプレイ要素を利用する。これらのコンソールは、個別のディスプレイ要素に近接して設けたボタンの形態をとることができ、あるいは、これらディスプレイ要素をボタンまたはキーキャップ自体に組み入れることができる。

【0337】典型的なサブマスタは、キューとともにストアされたタイミング・エフェクトをイネーブルにするかディスエーブルするための機構と、当該サブマスタにリコールされたキューでアクティブになったランプユニ

ットの可変パラメタ機能のうちの幾つかをフィルタリングする機構とを組み入れ、一方、本発明に係るコンソールは、当該本発明に係るコンソールでなければキューでアクティブにすることができない幾つかのランプユニットを、チャンネルごとにフィルタリングする機構も含むことができる。

【0338】キュー構造

本発明に係る制御コンソールには、フロントパネル機構が提供されるとともに、ボードレベル・キューと、ダイレクト・キューと、参照キューとを含むマルチレベル・キュー構造の使用をイネーブルにするオペレーティングシステム・プログラムを提供することができる。ダイレクト・キューは、1つのランプユニットの各機能に対して、絶対パラメタデータの形態で、可変パラメタデータを含むことができるデータセットである。しかし、ダイレクト・キューは、任意の機能に対して、1つ以上の絶対値に代えて、参照識別子も含むことができる。この参照識別子は、絶対パラメタデータを含む別のデータセットを指すインデックスとしてサーブするものである。

【0339】参照キューは別のデータセットであって、参照キューで当該データを単に修飾して、所定の参照キューにより構築された全ダイレクト・キューを新しい絶対パラメタデータに更新することができるように、ダイレクトキューで参照することができる絶対パラメタデータを含むデータセットである。

【0340】ボードレベル・キューはデータセットであって、所定の時点で、制御コンソールの種々のフロントパネル機構のステートを記述するデータセットであり、データセットには、例えば、どのダイレクト・キュー番号と、どのチェース・シーケンス識別子と、どのクロス・フェード・キューと、どのマトリクス・パッチ構成が、その時点で、どのソフト・サブマスタにロードされるかが含まれる。ボードレベル・キューをストアしリコールするフロントパネル機構が、ソフト・サブマスタに対して、ダイレクト・キューと、チェース・シーケンスと、等々から個々に提供される。参照キューをストアしリコールするためのフロントパネル機構には、ダイレクト・キューをストアしリコールするための機構から個々に提供される。あるいは、数値キーパッドのような幾つかの機構を、ダイレクト・キューおよび参照キュー調整つまみにより共有することができる。「ストア参照キュー(store reference cue)」用のボタンが「ストア・ダイレクト・キュー(store direct cue)」用のボタンから個々に提供される。

【0341】個々のサブマスタのために、「clear」または「thru」ステータスとともに、ボードレベル・キューをストアすることができる。「clear」ステータスは、ボードレベル・キューがリコールされると、任意の前のサブマスタ・エントリをサブマスタから除去する。「thru」ステータスは、ボードレベル・キューがリコー

ルされると、任意の前のサブマスタ・エントリをサブマスタに残す。ボードレベル・キューを制御リソース・ネットワークを介して伝送し、ボードレベル・キュー・アクティビティがネットワークに存在するか否かを各コンソールをリスン(listen)し、ボードレベル・キュー・アクティビティを、コンソールの前のステートにより解釈して、外部イベントをトリガするか、外部イベントによりトリガされる。別のコンソールのボードレベル・キュー・アクティビティに応答するように構成して、ネットワークのマルチプル制御コンソールを、当該ネットワークの複数のコンソールのうちの1つのコンソールから1つのボードレベル・キュー・リコールを行うことにより構成することができるようにすることができる。

【0342】トラッキング

1つ以上のランプユニットに対して同一のパラメタ値を有するダイレクト・キューの数値シーケンスに、トラッキング機能を適用することができる。例えば、各キューで、異なるパンおよびチルト値を指定している間、キュー10.01 ないし10.09 は、全て、1つ以上のランプユニットに対して同一の色を指定することができる。この例では、本発明に係るトラッキング機構によれば、シーケンス(10.01)の第1キューは、トラッキングされるパラメタ・データを含む「ソース・キュー(source cue)」である。キューの数値シーケンスをトラッキングするように色が機能され、その結果、任意の時点で、任意のキューに対して色が変更された場合、トラッキング機構に従うシーケンスの全てのキューで色が変更される。ユーザ選択可能オプションにより、現在キューから当該シーケンスのエンドにかけて適用されトラッキングされた機能が変更されたか否かが判定されるか、あるいは、当該シーケンスの1つのキューの変化が、元のソースキュー(10.01)からエンド(10.09)でにかけて適用された否かが判定される。ソースキューは、実際のパラメタ・データの代わりに、参照キューの識別子も含むことができる。

【0343】2つのモードのうちの一方のモードで、トラッキングを適用することができる。標準モードでは、全ての連続シーケンシャル・キューであって、当該シーケンスで、特定のランプユニット・パラメタを変化させないキューで、トラッキングが機能するが、当該シーケンスのその後のキューで、元のパラメタ値に当該パラメタが変化したとしても、当該パラメタが変化したとき、トラッキングを停止させる。オプション・モードでは、当該シーケンスのその後のキュー(このキューで、パラメタ値が元の値に戻される)に対して、トラッキングがレジュームする。標準またはオプション・モードが適用されたか否かを判定するため、ユーザ選択可能オプションを利用することができる。

【0344】データをトラッキングするための「ソース・キュー」であるチェース・キューの第1キューと、ソース・キューをトラッキングするチェースのその後の全

てのキューにより、トラッキングを、チェースにも適用することができる。よって、キューの任意またはランダムなシーケンスを確立することができる。というのは、キュー数値の任意のリストか、キュー数値の数値的にシーケンシャルなリストとして、チェースをプログラムすることができるからである。

【0345】以前には、シーケンシャル・キュー・リコール・オペレーションの間のタイムを制御する単一のチェース・レート値で、チェース・シーケンスが実行された。本発明に係る制御コンソールには、フロントパネル機構が提供されるとともに、チェース・シーケンスの各キューの間の可変タイミングの使用をイネーブルにするオペレーティングシステム・プログラムが提供される。これは、シーケンスの第1キューからその後のキューまでのキュー・リコール・オペレーションであって新しいチェース・レートが指定されるオペレーションに適用される第1チェース・レートの形態をとる。その後の全てのキューに対して新しいチェース・レートが確立されたか否かがユーザ選択可能オプションにより判定されるか、あるいは、(新しいチェース・キューの方が先行し(「例外」モード)、この新しいキューの後に、第1チェース・レートが再び適用される)キューにのみ、新しいチェース・レートが適用された否かが判定される。複数の新しいチェース・レートを単一のチェース・シーケンスに挿入することができる。標準モード変更または例外モード変更をオペレートするため、新しいチェース・レートをフラグ(flag)することができる。

【0346】タイミング制御

キューでアクティブになるランプユニットの個々の機能に対して、ダイレクト・キューはタイミング制御、速度制御、および/または、ウエイト・タイムを含むことができる。タイミング制御により、1つ以上の可変機能は、前のステートから現在のコマンド・ステートに、指定された時間に亘って、遷移させることができる。個別の時間は、異なる機能、例えば、パンと、チルトと、色と、ズーム・サイズと、エッジと、遮光板またはイメージ・ローテーションと、光度と、等々のような機能に対して可能である。速度制御により、1つ以上の可変機能は、前のステートから現在のコマンド・ステートに、指定された変化レートで変化させる。個々の速度は異なる機能で可能である。ウエイト・タイムにより、1つ以上の可変機能は、前のステートから現在のコマンド・ステートへの遷移を遅延させる。個々のウエイト・タイムは異なる機能で可能である。ダイレクト・キューは種々の機能に対して複数のウエイト・タイムを持つことができる。これら種々の機能は、プログラムされたウエイト・タイムの後に、前のステートから現在コマンドされているステートへの遷移が、プログラムされたタイミングまたは速度制御によりモジュレートされるように、タイミング制御または速度制御に結合されている。

【0347】各キューとともにストアすることができるタイミング制御データは、簡単な「one time for all parameters(全てのパラメタに対して1度に)」を含むことができるか、幾つかのブロード・クラスのパラメタ、例えば、光度と、フォーカスと、カラーと、ビーム・パラメタに対して、それぞれ、個別のタイムを含むことができる。本発明に係る制御コンソールは、図34に示すようなフロントパネル機構をさらに含み、ランプユニットの各可変パラメタに対して、個別のタイミング制御値のプログラミングをイネーブルにするオペレーティングシステム・プログラムをさらに含む。例えば、特定のランプユニットの色制御機能が3つのモータライズ・カラーフィルタ・ドライブにより制御された場合には、当該ランプユニットに対する特定のキューとともにストアされたタイミング制御データは、3つのカラーフィルタ・ドライブに対して、それぞれ、個別の実行タイムか、速度制御値を含むことができる。あるいはまた、単一のタイミング制御値は3つのカラーフィルタ・ドライブに等しく適用することができる。全ての機能に対する単一のタイムであって、そのデフォルト値がゼロ・タイムかフルスピードであるタイムは、各ランプユニット機能に対して最初にエンタされるように、階層構造をタイミング制御値に課することができる。

【0348】その後、ブロードクラス機能、例えば、色またはフォーカスのような機能に対して、タイミング値をエンタすることにより、当該ブロードクラスに含まれる具体的な全ての機能に対して、単一のタイミング値を書き換える。例えば、影響を受けるランプユニットの各カラーフィルタ・ドライブに対して、全ての機能に対して単一タイムに亘って課されるカラータイムにより、新しいタイミング値がエンタされる。その後、当該ランプユニットのカラーフィルタ・ドライブのうちの1つのドライブのみに対して、第3タイミング値をエンタして、カラータイミングを修正することができ、他のカラーフィルタが、ブロードクラス機能の「color」に対してエンタされるタイミング値により制御されている。ユーザ選択可能オプションにより、ブロードクラス機能に対してエンタされた新しいタイミング値が具体的なパラメタ・ドライブに対する全てのタイミング値を書き換えた（「reset」モード）か否かが判定されるか、あるいは、新しいタイミング値が、具体的なタイミング値で、プログラムされていなかったパラメタ・ドライブにのみ伝播される（「fill-in」モード）か否かが判定される。

【0349】1つのキューで、ランプユニットのブロードクラス機能に対して、それぞれ、ウエイトタイムをエンタすることができるか、あるいは、各ブロードクラス機能に対して具体的な各パラメタ・ドライブに対して、ウエイト・タイムをエンタすることができる。タイミング値に関して上述したのと同じ階層構造により、ウエ

イト・タイムをプログラムすることができ、上述したような「reset」または「fill-in」モードとともに、ウエイトタイムをインプリメントすることができる。

【0350】前のイベントを完了した後、ボードレベル・キューが自動的に開始するように、オート・フォロー・モードをボードレベル・キューに適用することができる。このようにして、イベントの複雑なシーケンスの実行を簡単にするため、あるボードレベル・キューは別のあるボードレベル・キューを自動的にフォローすることができる。シーケンシャルな数値を有するダイレクト・キューが自動的にリコールされ、任意のウエイト・タイムと、キューと共にストアされたタイミング制御値とを監視するように、オート・フォロー・モードをダイレクト・キューにも適用することができ、その結果、オート・フォローは、初期のキューがどのようにロードされたかに関係なく、常に、生じられる。

【0351】インテリジェント照明アシスタント

本発明に係る分散制御システムは、制御システム要素と、本システムが人間のオペレータに対して「インテリジェント照明アシスタント」として応答することができるオペレーティングシステム・プログラムとを含むことができる。制御資源ネットワークに接続された汎用コンピュータにインプリメントされた推論エンジンは、音声認識モジュールまたは他の入力デバイスからのコマンド入力を受信する。音声認識デバイスおよび技術は、当業者に周知であり、例えば、ProAudio16 サウンドカードと、Media Vision of Freemont, Californiaが市販するExecuVoiceソフトウェアがある。推論エンジンは、ショーファイル（照明システムの3次元モデルと、本明細書でモデル化されたランプユニットに対するパラメタ・データセットのデータベースとを含む）を参照してコマンドを評価する。このショーファイルは制御資源ネットワークに接続されているシステムデータ・レポジトリに保持されている。一般化された命令入力をパラメタ調整データ出力に変換するため、この推論エンジンは、図35に図示された方法を利用してオペレートされる。この方法には、一般化された主観的なコマンド入力、例えば、「the upstage lamp units in red change to blue (赤から青になったステージ奥のランプユニット)」というフレーズのような入力を、ランプユニット・アドレスに変換するためのルールのセットを適用して、マニュアル制御およびパラメタ調整値を選択し、ランプユニットをドライブして、必要な属性またはパラメタを有する光ビームを生成する。推論エンジンは、本発明に係る方法により、そのコマンド入力を分析して、どのランプユニットがそのコマンドにより影響を受けるかを判定し、このコマンドがこれらランプユニットにどのように影響を与えるかも判定する。最後に、この制御コンソールは適正なシステム・コマンドをコンポーズ(compose)し、このコマンドを高速制御資源ネットワーク600を介して

伝送する。

【0352】ユーザ選択可能オプションにより、影響を受けたランプユニットがこのコマンド入力に応答する前に、個別の「go」コマンド入力が必要か否かが判定され、あるいは、結果として行う調整が、制御システムがそのコマンドを解釈すると直ちに行われる。

【0353】本発明に係る推論エンジンは、このシステムデータ・レポジトリに保持された3次元モデルに含まれるオブジェクトの相対的なロケーションを記述する共通言語表現(common-language expressions)を区別することができる。結果として得られるパラメタ調整値は、少なくとも、特化された意味を当該産業でほとんど一般的によく理解されるように、ワードと、フレーズと、照明産業で特化された意味を有する他の表現とが、推論エンジンにより解釈される。1つの表現が2つ以上の意味を持った場合に予測されない結果を避けるため、他のワードか、フレーズか、あるいは表現には、特化された意味が必要に応じて与えられる。本発明に係る推論エンジンは、そのため、産業上の「ジャーゴン」を正確に解釈し、特化された意味を有するより多くの表現を含むため、そのジャーゴンを拡張することができる。その入力を評価してチャンネル選択要件を判定するために用いられる知識は、共通の知識用語として、例えば、「first(最初)」と、「last(最後)」と、「odd(奇数)」と、「even(偶数)」と、等々の用語を含み、技術上のジャーゴンとして、「first electric」と、「torm」と、等々の用語を含み、3次元モデルから検索された知識として、「upstage(ステージ奥)」と、「stage right(ステージ右)」と、「back truss(バックトラス)」と、等々の用語を含み、現在ランプユニット属性として、「green lights(緑ライト)」と、「lights on the singer(歌手にライトを当てる)」と、等々の用語を含む。その入力を評価してパラメタ調整要件を判定するのに用いられる知識は、コンソール構成として、「preset color(プリセット・カラー)」と、「preset focus(プリセット・フォーカス)」と、等々を含み、数値パラメタ・エントリとして、「at intensity 75(光度を75で)」と、「in gobo 3(遮光板3で)」と、等々を含み、3次元モデルから検索される知識として、「at the singer(歌手を狙って)」と、「two feet upstage(2フィートステージ奥に)」と、等々を含み、現在ランプユニット属性として、「the same green as those other lamp units(当該他のランプと同じように緑にする)」と、等々を含む。本発明に係る推論エンジンは、システム・データ・レポジトリと対話を行い、制御資源ネットワークと関係付けをしたグラフィック・ディスプレイと対話を行って、オンラインおよびオフラインで、コマンド入力表現の結果をディスプレイする。人間のオペレータは、コマンド入力インタフェース、例えば、音声認識モジュールと、汎用コンピュータにインプリメントされた

推論エンジンと、照明システムの3次元モデルのようなコマンド入力インタフェースを備えたインテリジェント照明アシスタントのみを用いて照明システムをプログラムすることができる。当該ランプシステムのうちの残りの休眠しているランプユニットのアクティビティをシミュレートし、一方、入力インタフェースと推論エンジンは、オフライン・プログラミングに提供するため、制御コンソールの代わりに用いられる。コマンド入力表現の結果は、グラフィック・ディスプレイにディスプレイされ、上述したような、フォト・リアリスティック・ビューか、ワーキング・ビューか、チャンネル・ユーセジ・ディスプレイを表示する。あるいはまた、異なるディスプレイに、異なるビューを表示して、複数グラフィック・ディスプレイを利用することができる。「インテリジェント照明アシスタント」が人間のオペレータに話することができるようにするため、ボイス・シンセサイザ・モジュールを本システムに含めることができる。

【0354】当該推論エンジンが解決することができない曖昧さが存在する場合、問いを、グラフィック・ディスプレイにテキストで表示するか、あるいは、電氣的にシンセサイズされた音声を生成するかのいずれかで、人間のオペレータに発する。例えば、「lights on the singer(歌手にライトを当てる)」という表現を含む入力を評価するため、当該システムは「Which singer do you mean? (どの歌手か?)」という問いを発することができる。この問いを、ボイス・シンセサイザ・モジュールを利用して、聞き取れるように通信を行うか、あるいは、その問いを、グラフィック・ディスプレイのウインドウに、テキストで表示することができる。別の例としては、入力表現として、「the same green as those other lamp units(当該他のランプユニットと同じように緑にする)」を含む場合は、本システムは「Which other lamp units do you mean? (他のランプユニットはどれか?)」という問いを発することができる。

【図面の簡単な説明】

【図1】本発明を具現化することができるタイプのコンピュータ制御照明システムの斜視図である。

【図2】本発明を具現化することができるタイプの照明システムであって、制御コンソールと種々のランプユニットの間での通信を示し、同様に、ステージ装置の他のアイテムを示すブロック図である。

【図3】本発明を具現化することができるタイプの照明システムの制御コンソール用のフロント・パネルを示すブロック図である。

【図4】制御コンソールの一部であることが可能なタイプの電気サブシステムを示すブロック図である。

【図5】ランプユニットのランプ・プロセッサ・システムの電気ブロック図である。

【図6】ランプユニットのステップ制御システムを示すブロック図である。

【図 7】ランプユニットのステッパ・モータとともに用いられるインデックス・センサ・システムを示すブロック図である。

【図 8】ランプユニット内モータの回転速度制御およびポジション監視を含むサーボ・フィードバック制御を示すブロック図である。

【図 9】図 2 に示すリピータを詳細に示す略図である。

【図 10】多くのセンシング、通信、他のオペレーション制御プログラムとをステップ処理するメインシーケンサを含む制御コンソールでのプログラムのオペレーションを説明するための流れ図である。

【図 11】本発明を具現化することができるタイプの照明システムのオペレーションを行う制御コンソールで利用される追加のプログラムを示す流れ図である。

【図 12】ランプユニットを初期化してオペレーションを開始するためにランプユニットで行われる個々のステップを示す流れ図（その 1）である。

【図 13】ランプユニットを初期化してオペレーションを開始するためにランプユニットで行われる個々のステップを示す流れ図（その 2）である。

【図 14】コマンド受信ユニットをステップ処理するメインシーケンサ・プログラムと、一連のテスト・プログラムとを含むランプユニットのプロセッサでのプログラムの基本的なオペレーションを示す流れ図である。

【図 15】パラメタ制御コマンドを受信し、これらパラメタ制御コマンドを処理する流れ図であって、光ビームにパラメタの選択されたセットを持たせるため、ランプユニット内のメカニズムにより行われる物理オペレーションをディレクトするランプ・プロセッサ内で行われるオペレーションを示す流れ図である。

【図 16】図 9 に示すリピータのブロック図である。

【図 17】マスタ/スレーブ関係にある 2 つの信号分配ラックの相互接続を示すブロック図である。

【図 18】本発明を具現化したタイプの照明システムを示すブロック図であって、マスタ（コンソール）リピータと、スレーブ（トランク）リピータと、該リピータとランプユニットに接続してある種々のトラス・リピータとを示すブロック図である。

【図 19】既存のブロードキャスト・ネットワークを示す照明システムのブロック図である。

【図 20】既存の応答ネットワークを示す照明システムのブロック図である。

【図 21】本発明を具現化したタイプの改良したリピータのブロック図である。

【図 22】本発明を具現化したタイプの別の改良したリピータのブロック図である。

【図 23】本発明を具現化したタイプの「スマート」リピータのブロック図である。

【図 24】本発明を具現化することができるタイプの改良したブロードキャスト・ネットワークを示す照明システムのブロック図である。

【図 25】本発明を具現化することができるタイプの改良した応答ネットワークを示す照明システムのブロック図である。

【図 26】モジュラ・コントローラ・メインフレームの一部を形成する電気サブシステムのブロック図である。

【図 27】モジュラ・コンソール・システムのブロック図である。

【図 28】分散制御システムを示すブロック図である。

【図 29】制御コンソール用の典型的なアーキテクチャを示す図である。

【図 30】汎用コンピュータ用の典型的なアーキテクチャを示す図である。

【図 31】負荷インタフェース・モジュール用の典型的なアーキテクチャを示す図である。

【図 32】負荷インタフェース・モジュール用の典型的なアーキテクチャを示す図である。

【図 33】ソフト・サブマスタを示す図である。

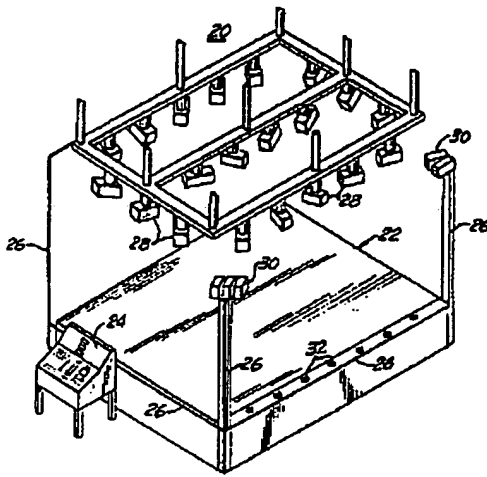
【図 34】種々のフロントパネル機構を示す図である。

【図 35】一般化された命令入力をパラメタ調整値データ入力に変換する方法を示す図である。

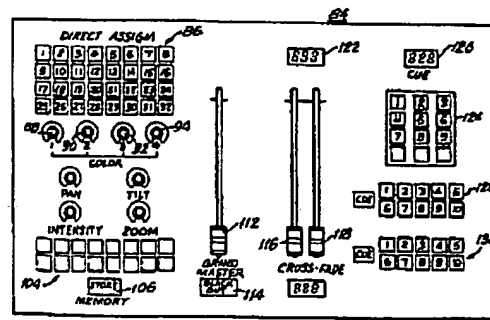
【符号の説明】

- 20 システム
- 22 ステージ
- 24, 82, 84 制御コンソール
- 26 データリンク
- 28 ランプユニット
- 30 ランプ
- 32 フロアランプ
- 42, 44, 46, 48, 50 ランプユニット
- 52, 54, 56, 58 リピータ
- 60 調光器
- 62, 64 制御信号コンバータ
- 66 チェインホイスト・モータ
- 68 エアキャノン
- 70 特殊効果プロジェクタ
- 80 両方向バス
- 178 ランプ・プロセッサ・システム
- 190 エンコーダ/デコーダ
- 200 マイクロプロセッサ
- 202 RAMおよびEPROMメモリ
- 218, 220 デコーダ
- 246 マルチプロトコル通信制御装置
- 254 パラメタ・ドライブ回路
- 278, 296 パラメタ制御回路

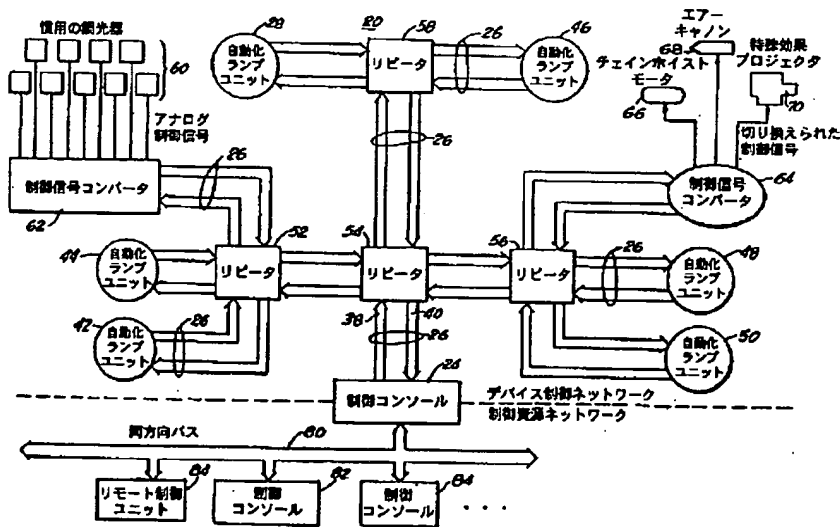
【図1】



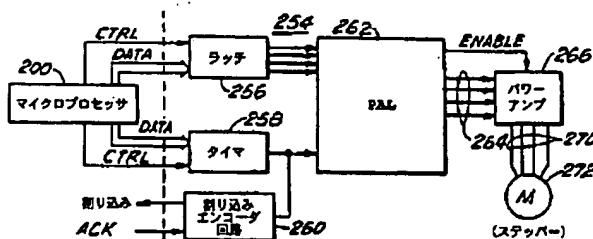
【図3】



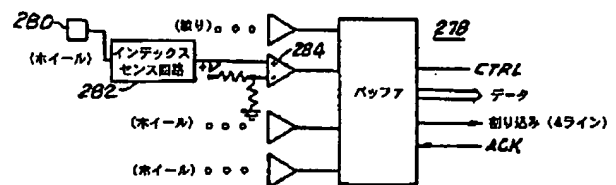
【図2】



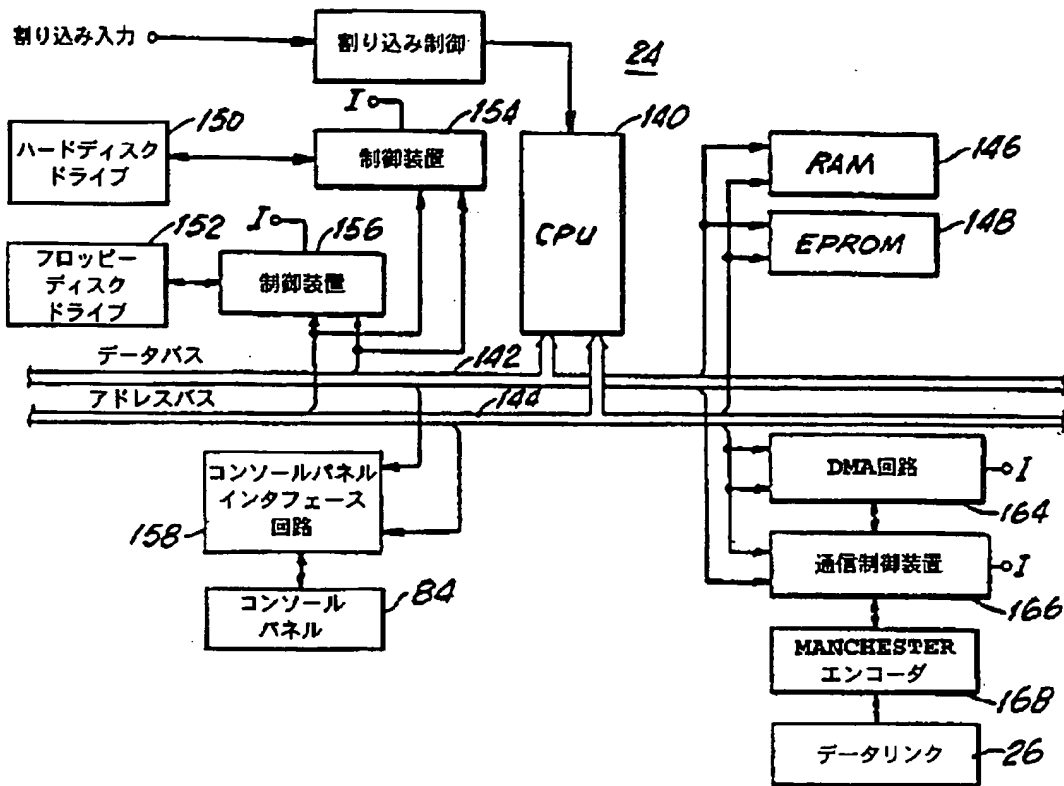
【図6】



【図7】

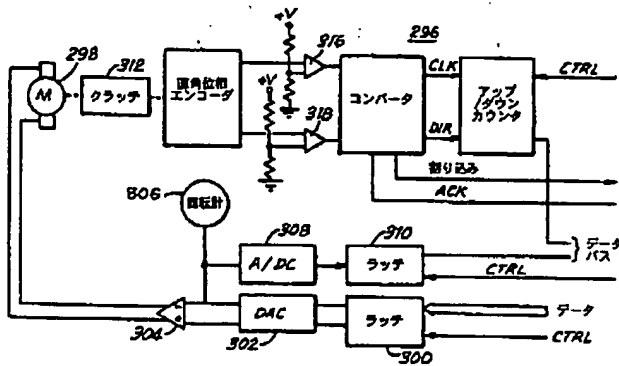


【図4】

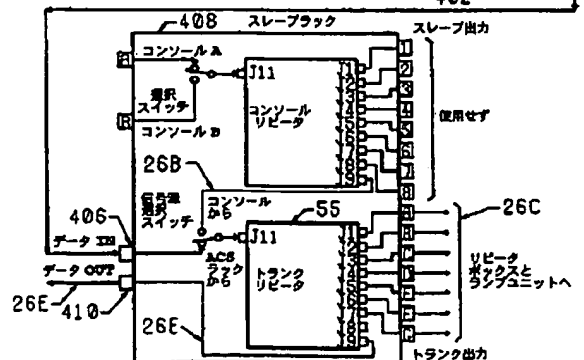
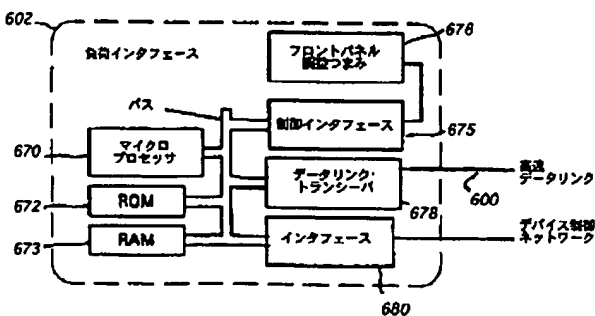
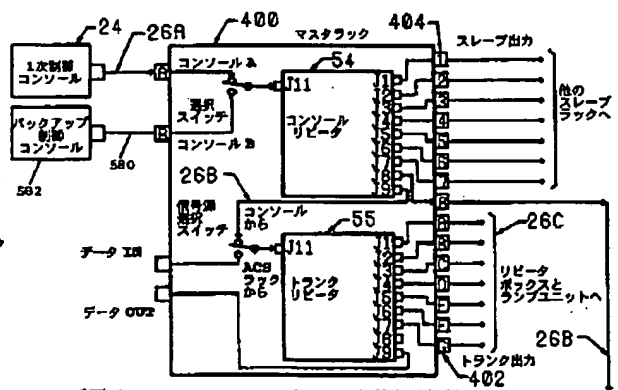


【図8】

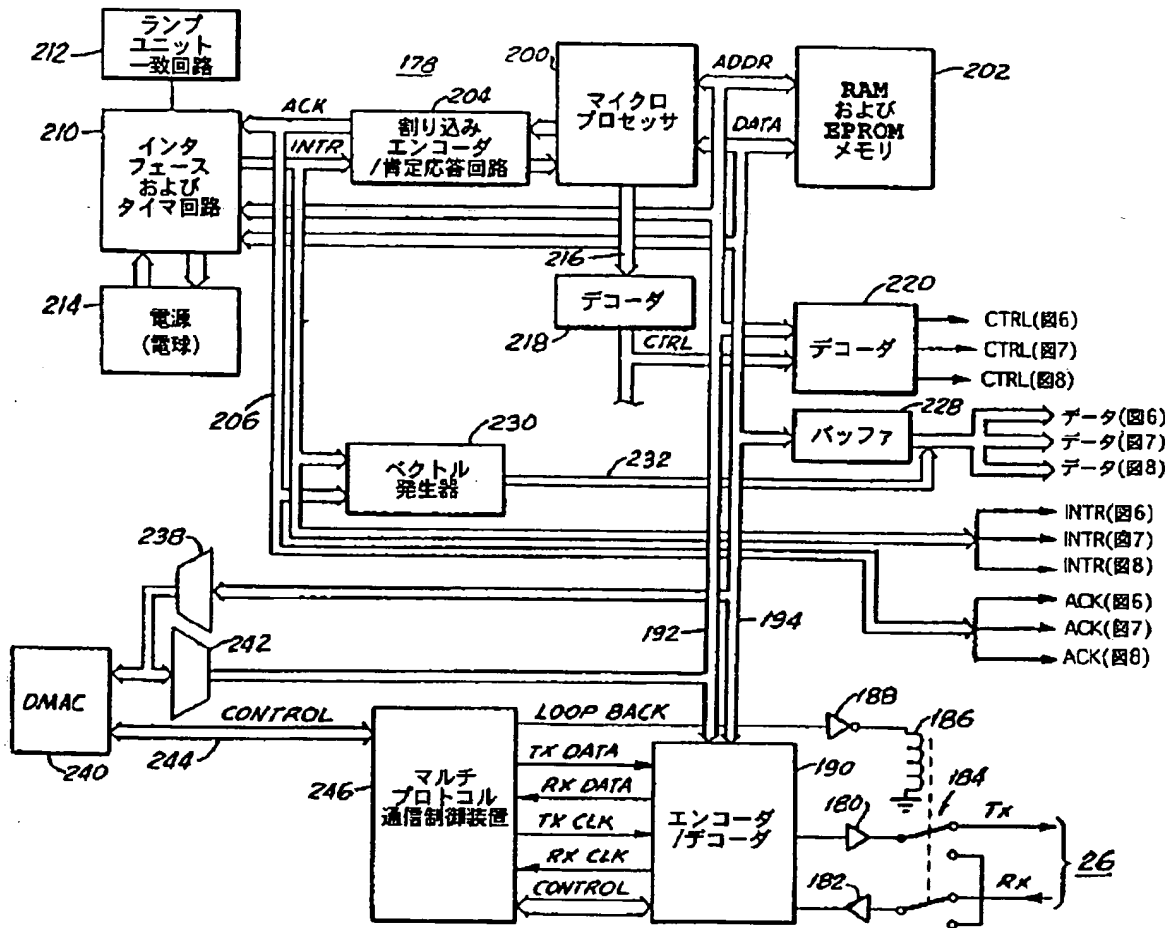
【図17】



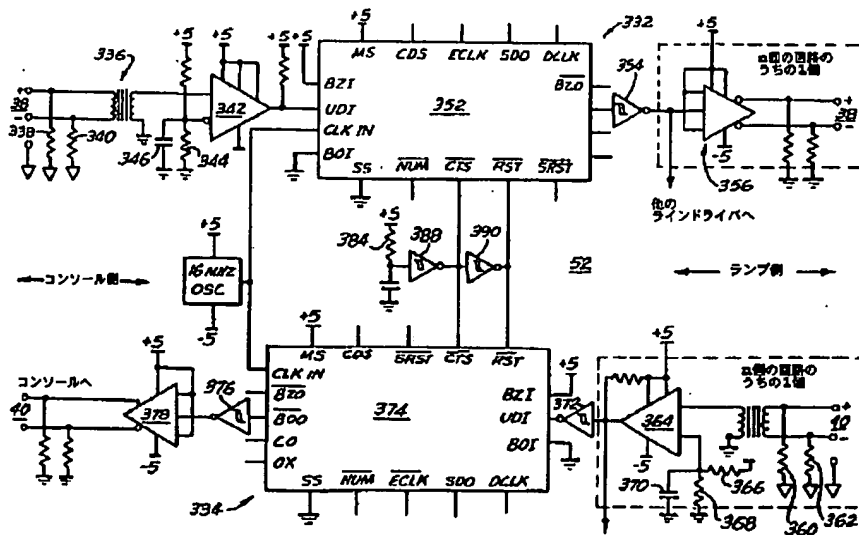
【図31】



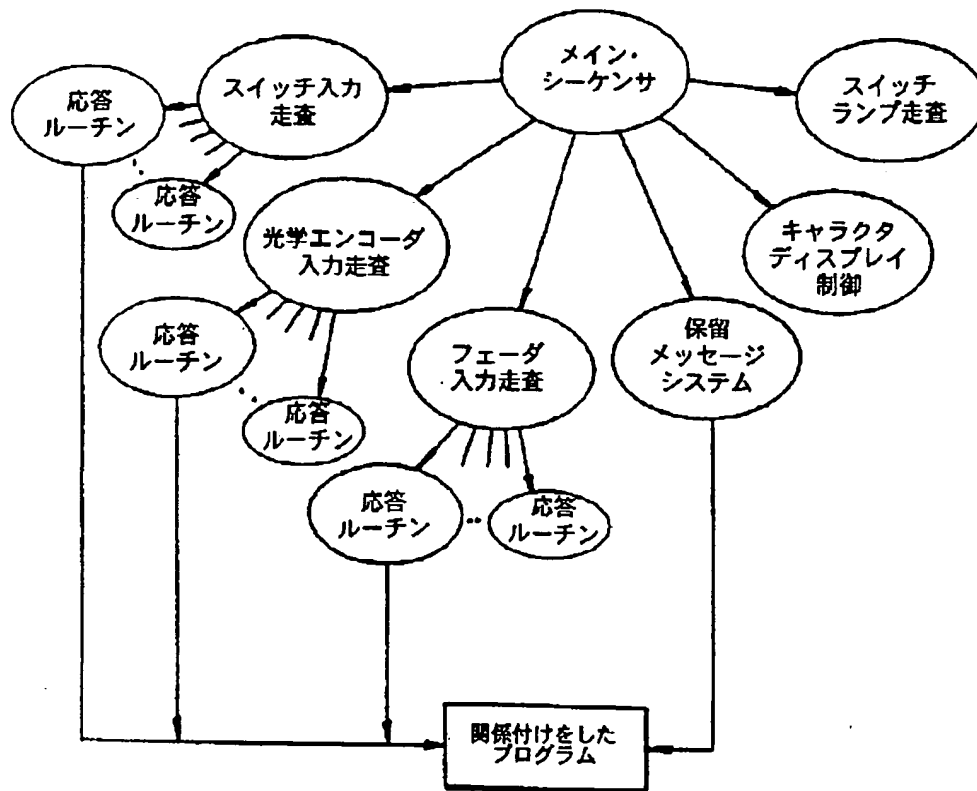
【図5】



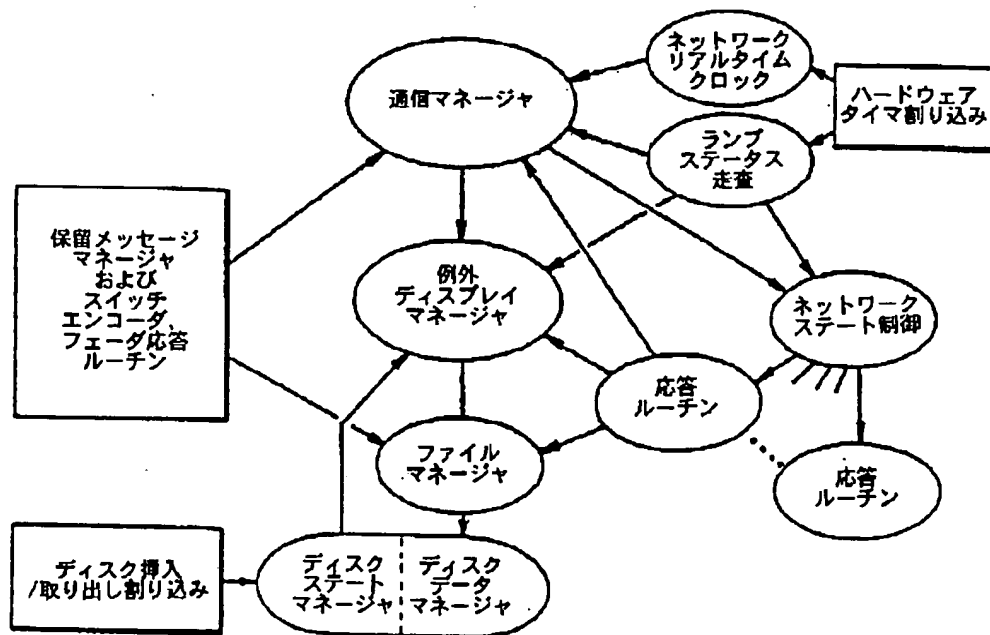
【図9】



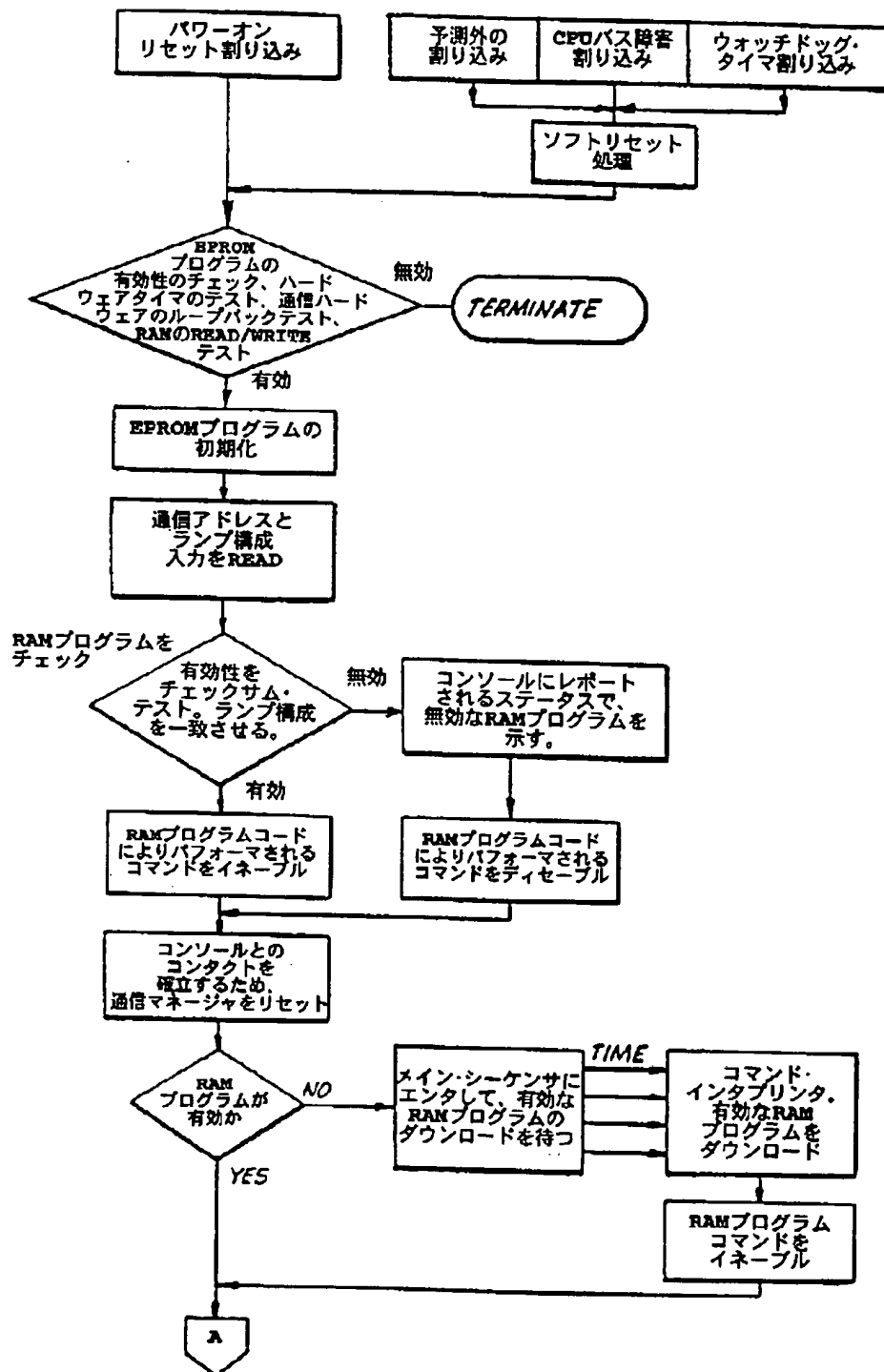
【図10】



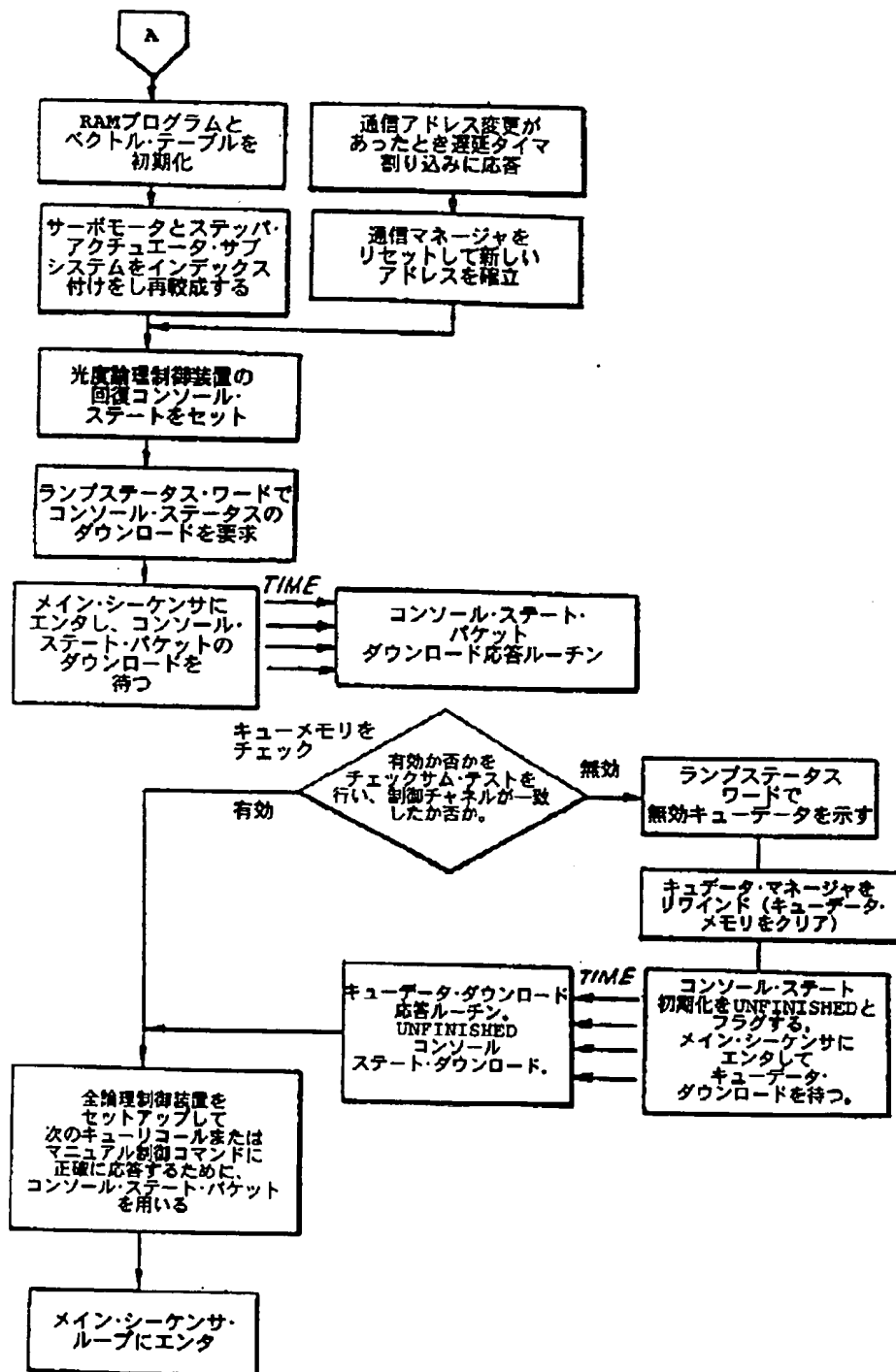
【図11】



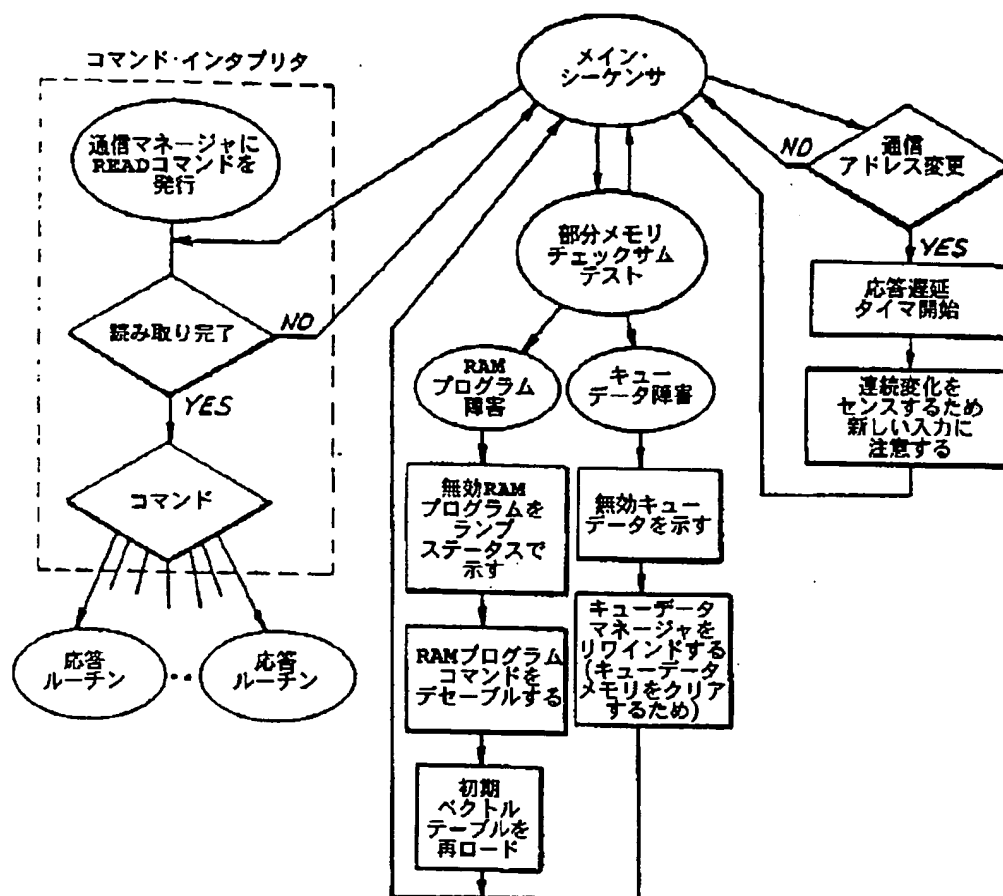
【図12】



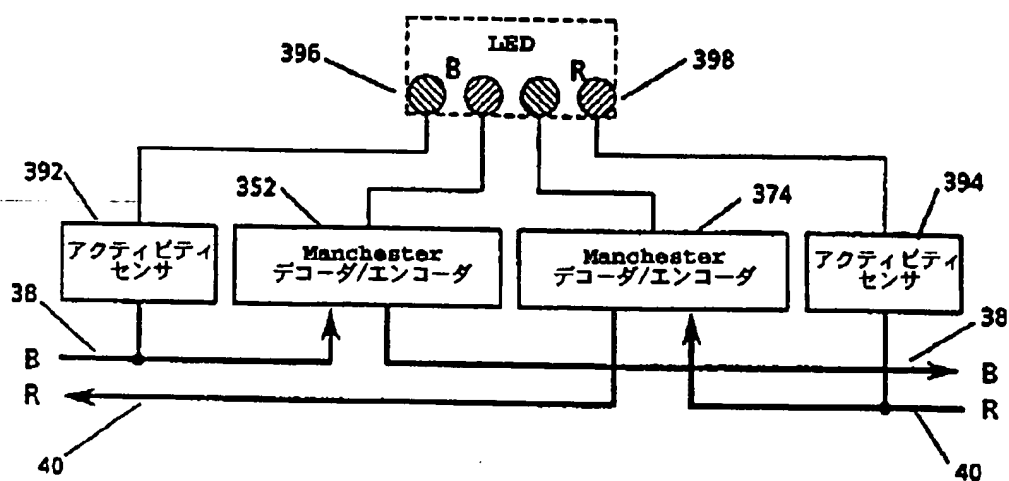
【図13】



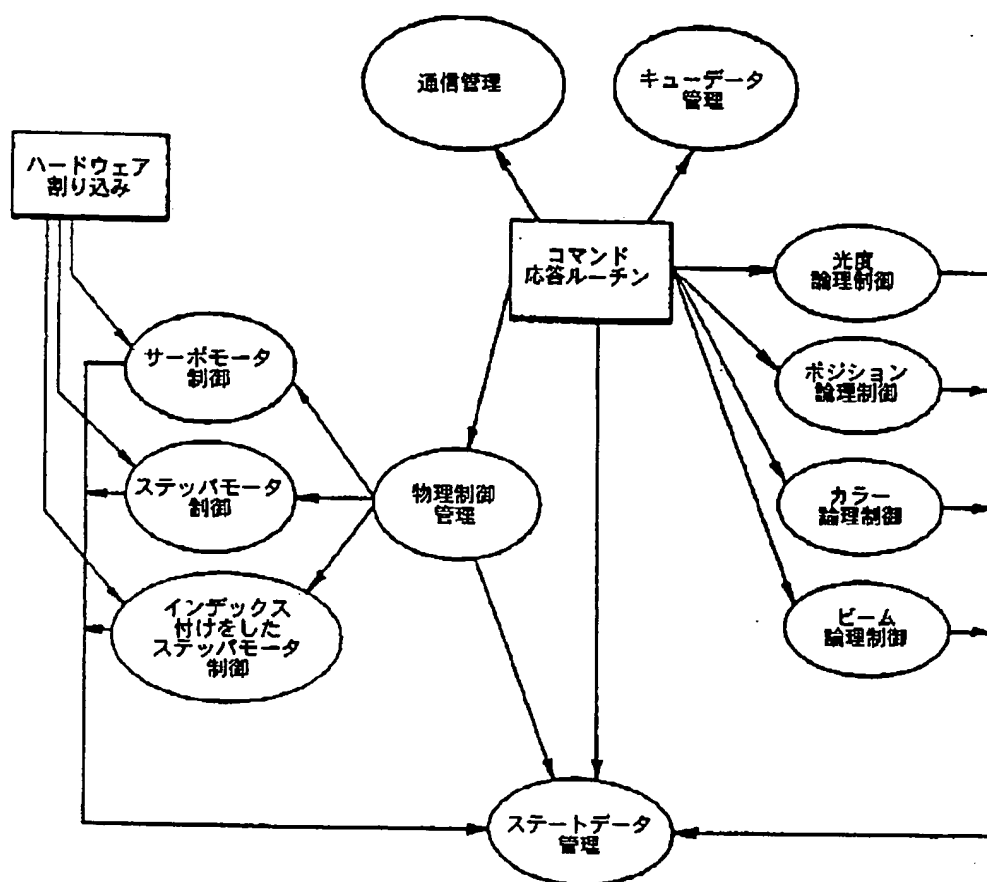
【図14】



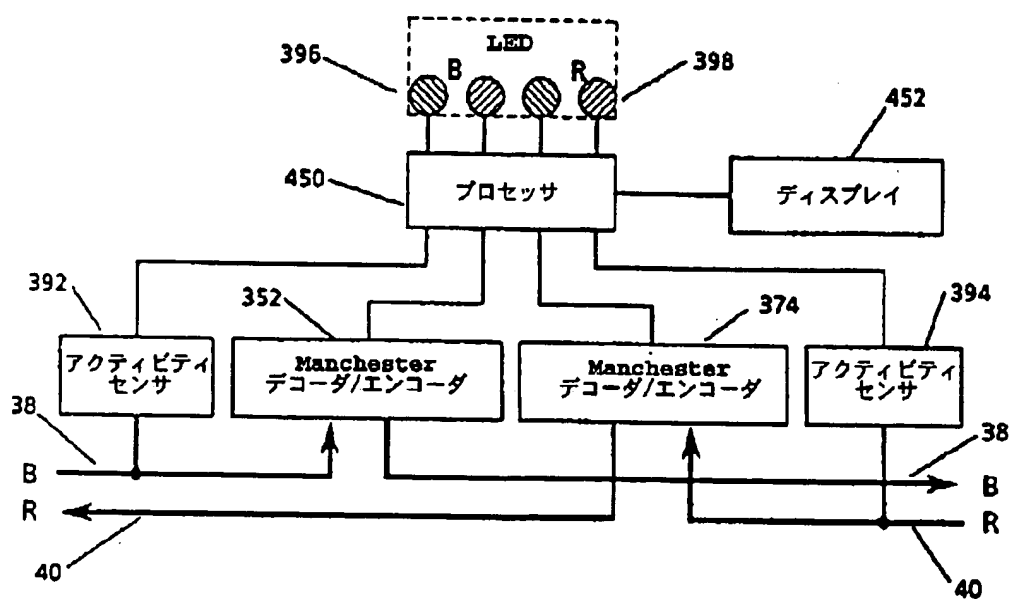
【図16】



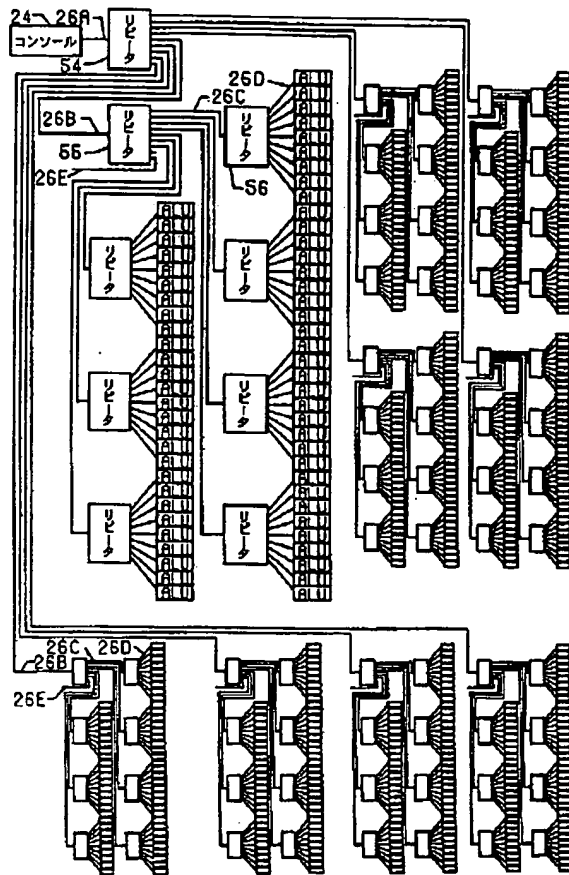
【図15】



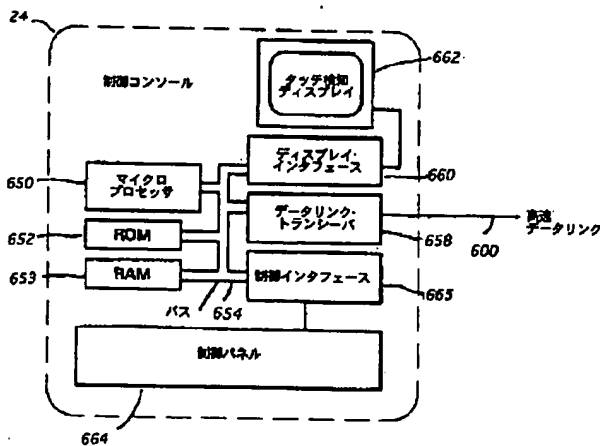
【図21】



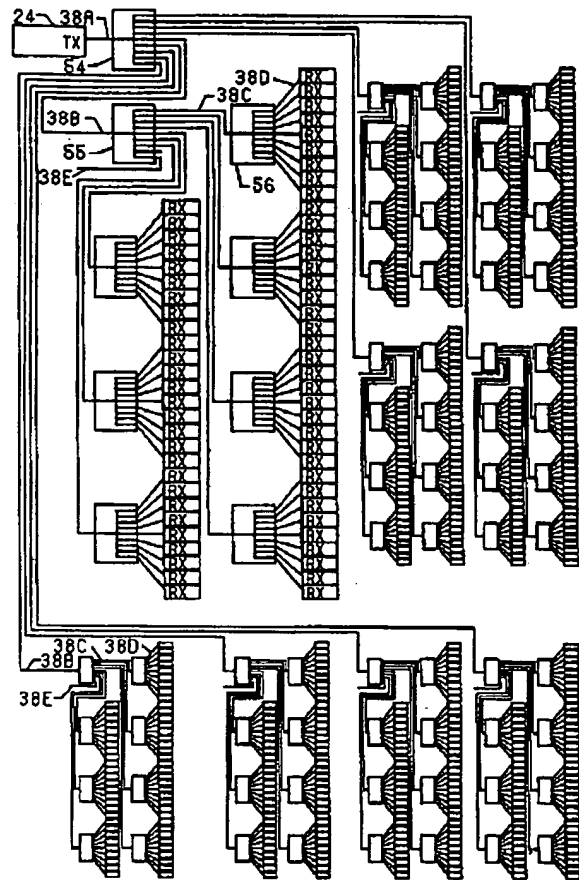
【図18】



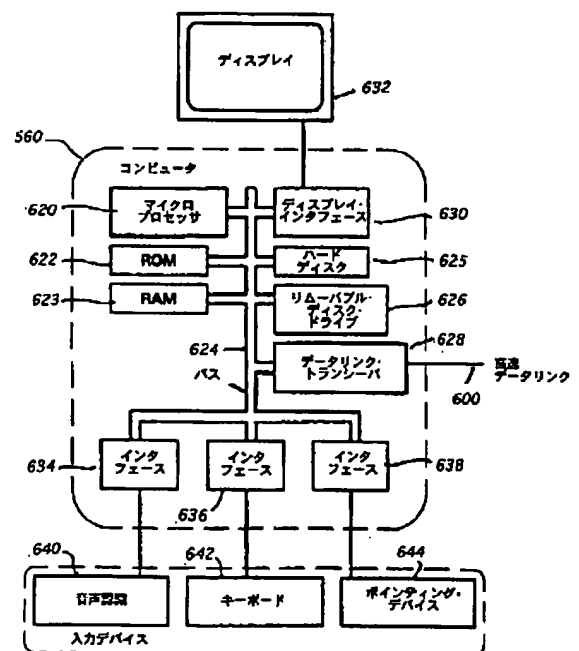
【図29】



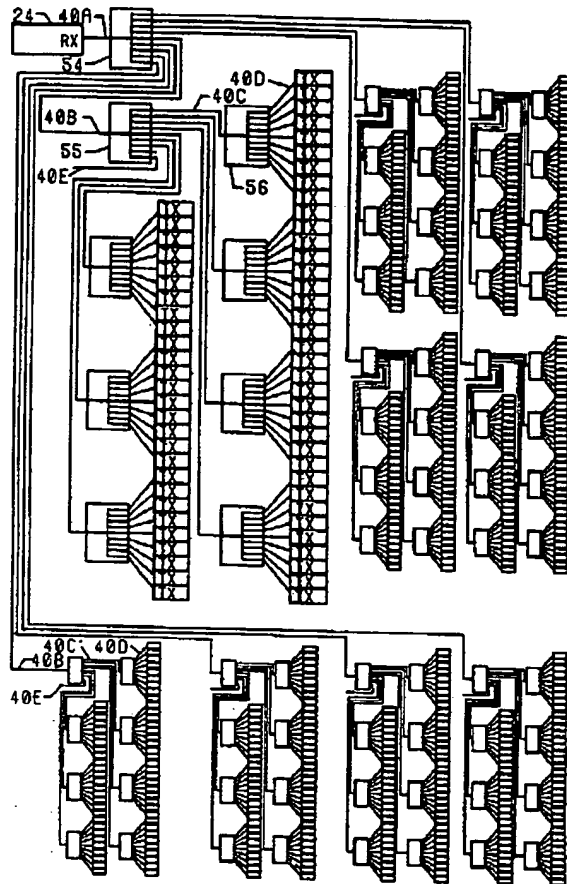
【図19】



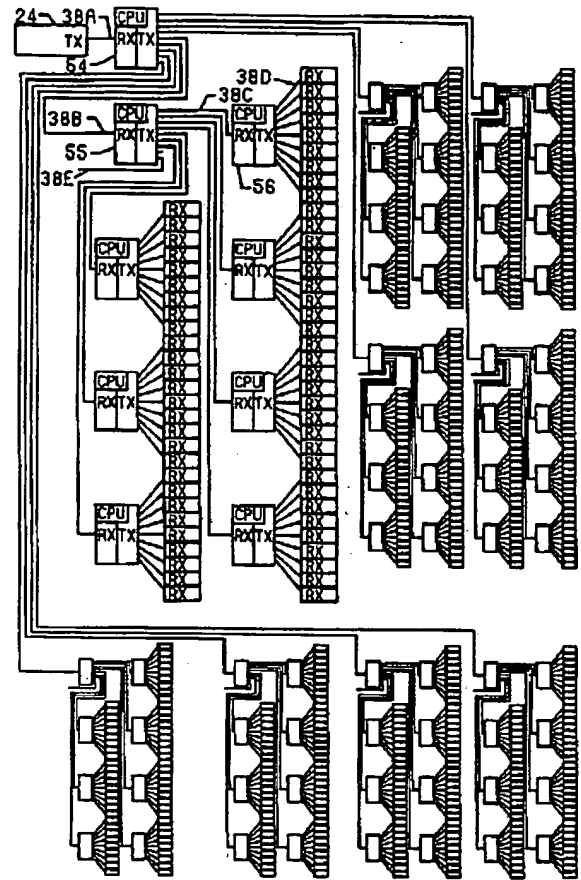
【図30】



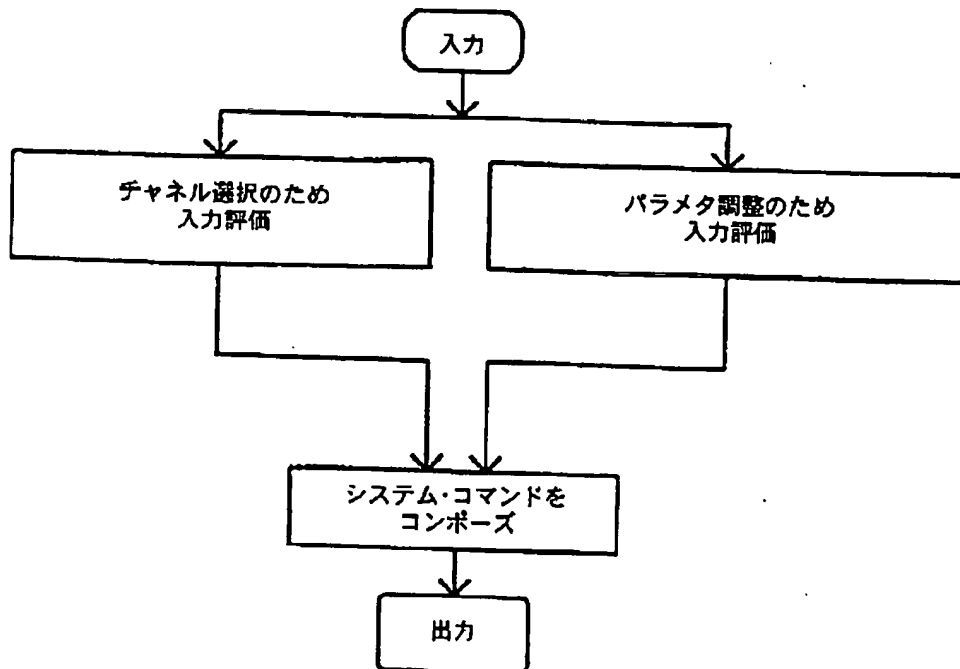
【図20】



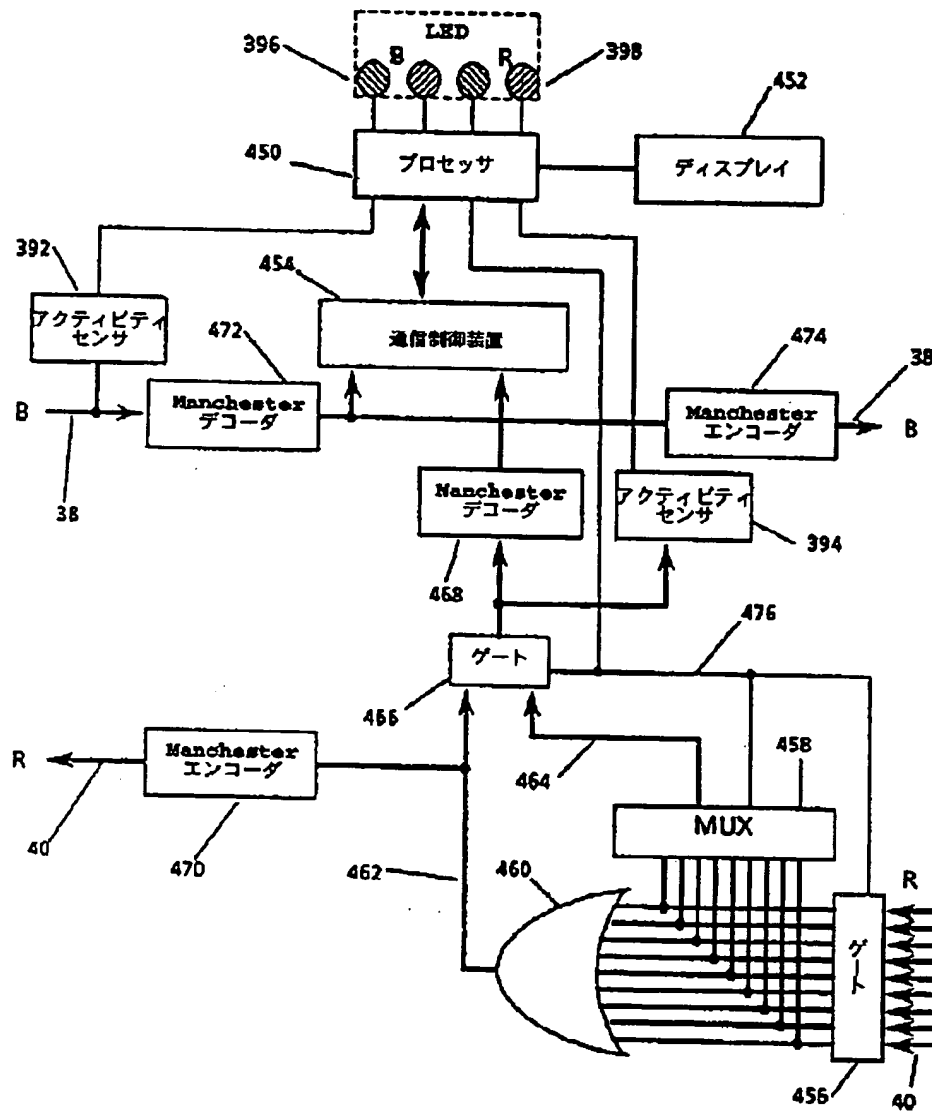
【図24】



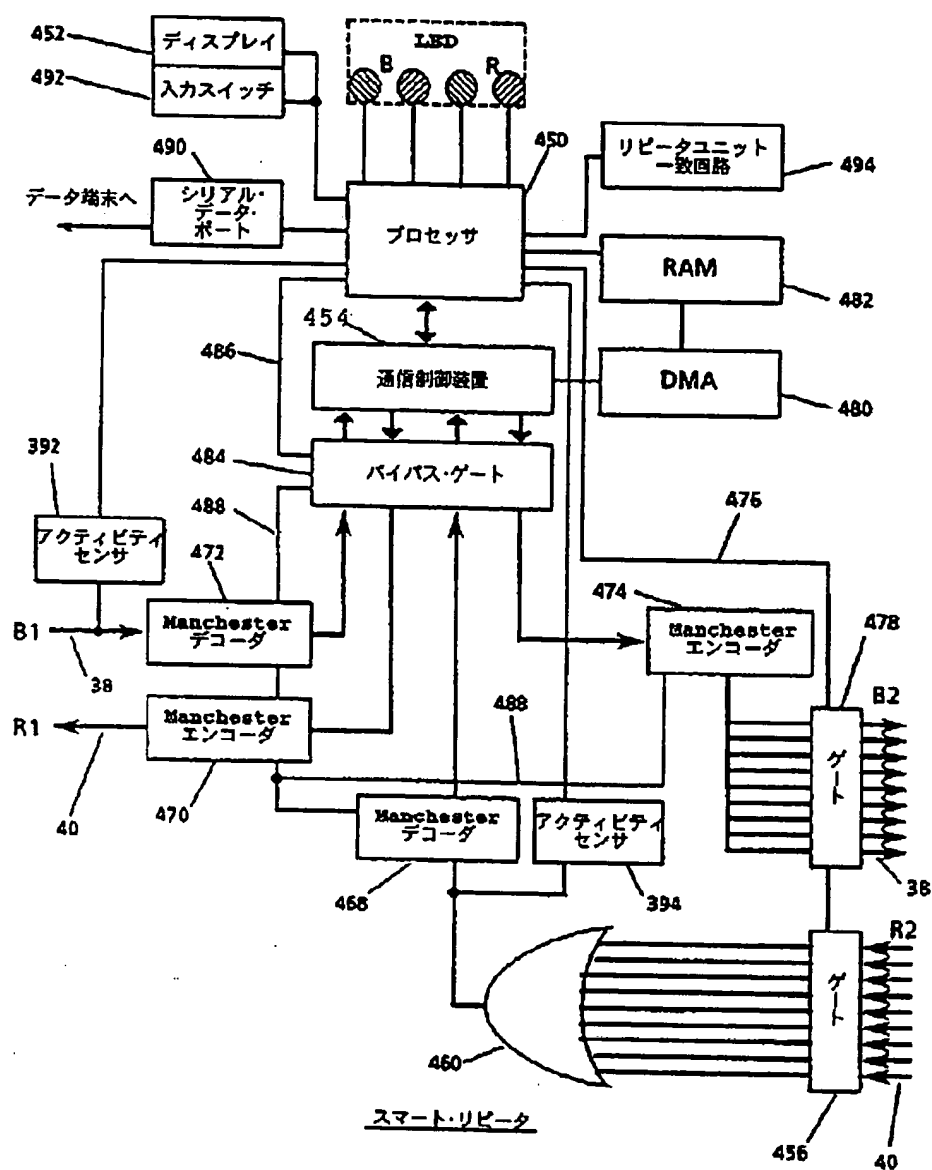
【図35】



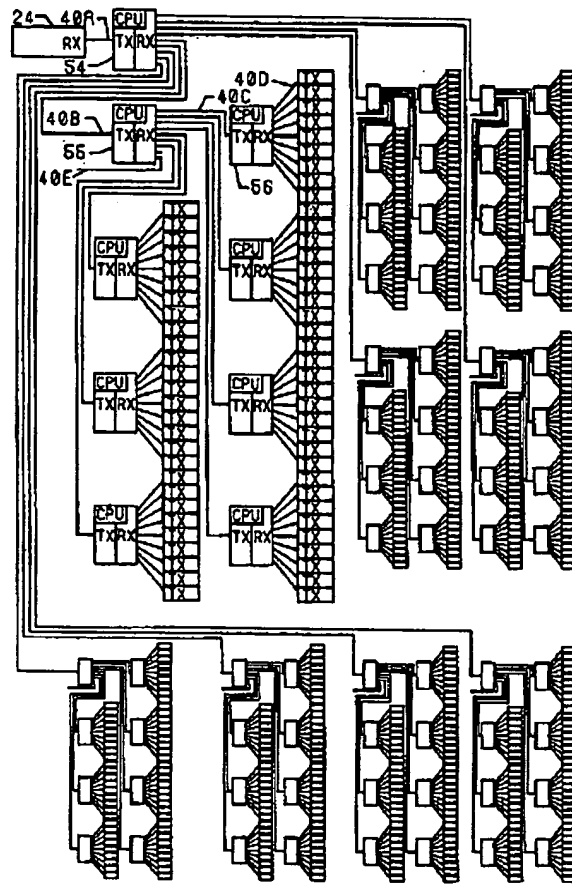
【図22】



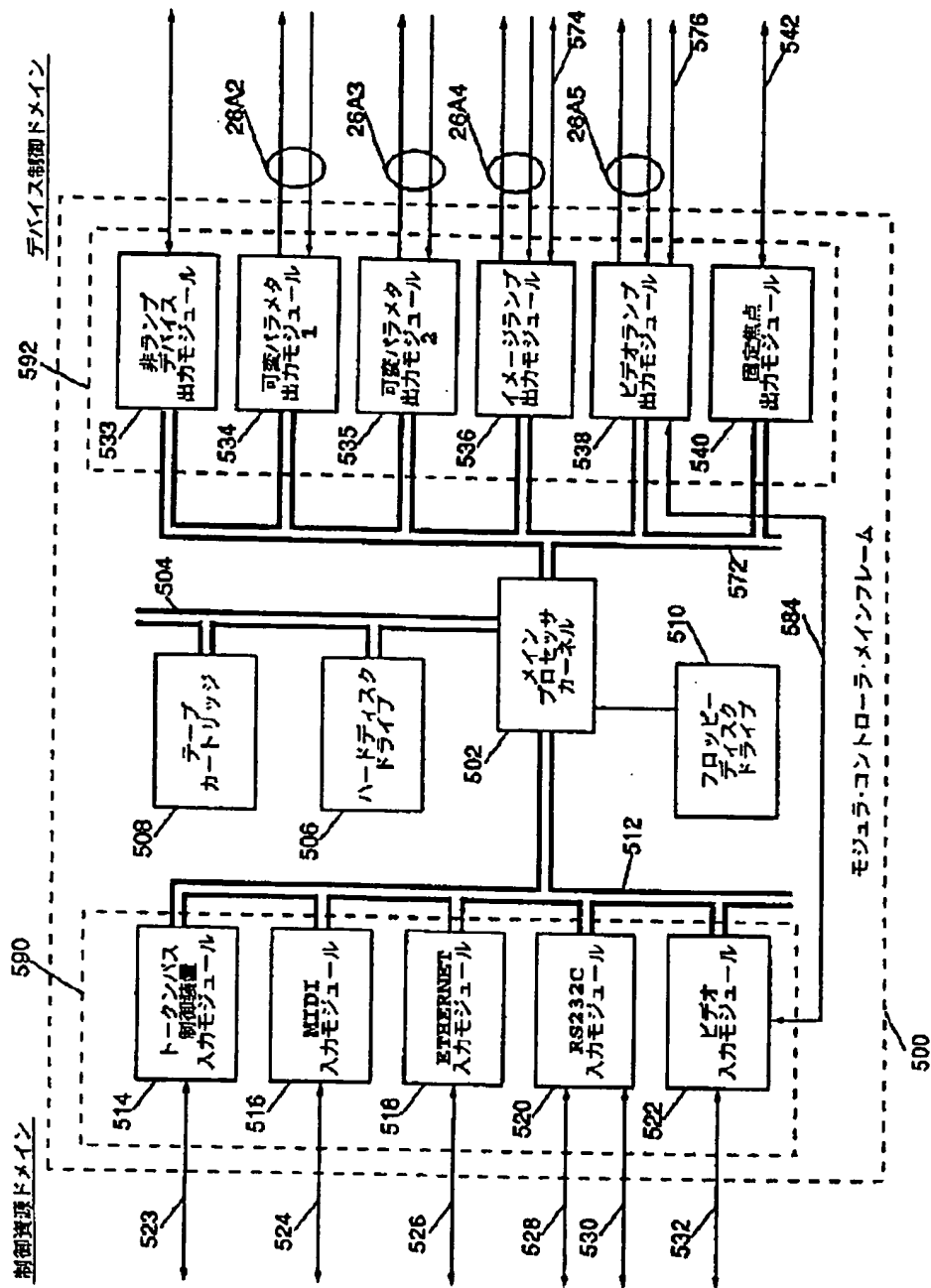
-63-



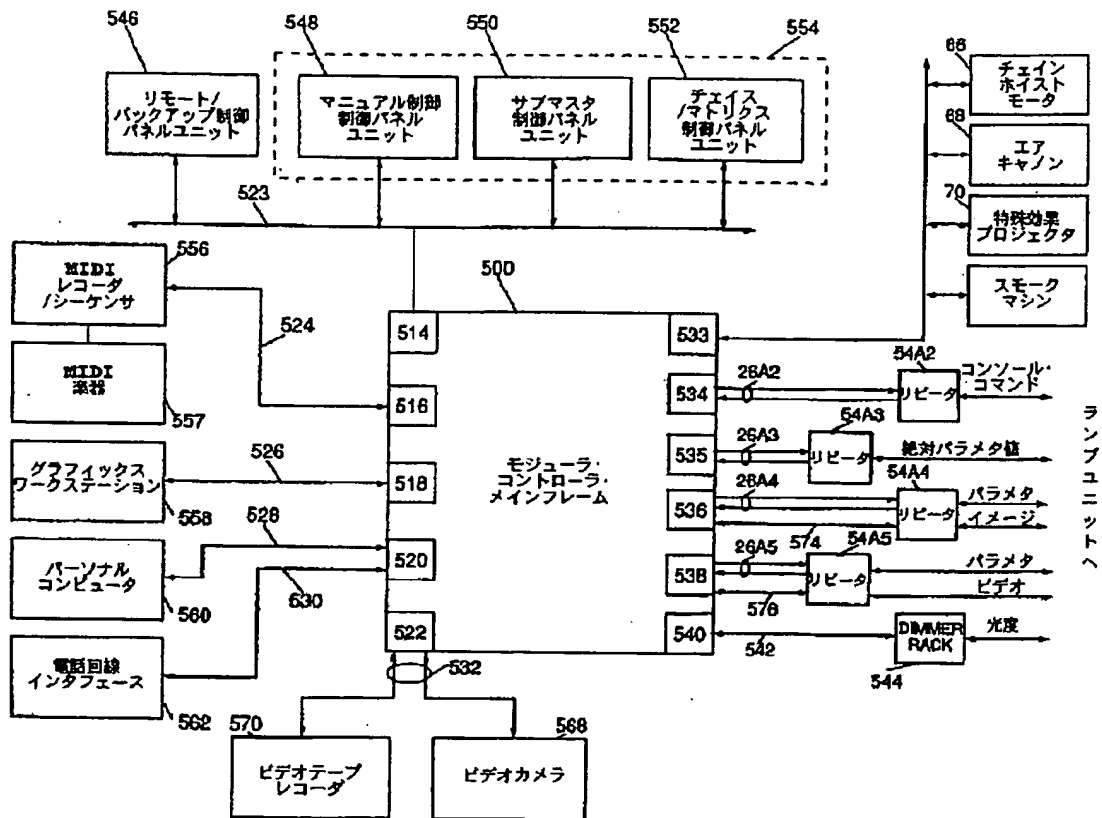
【図 25】



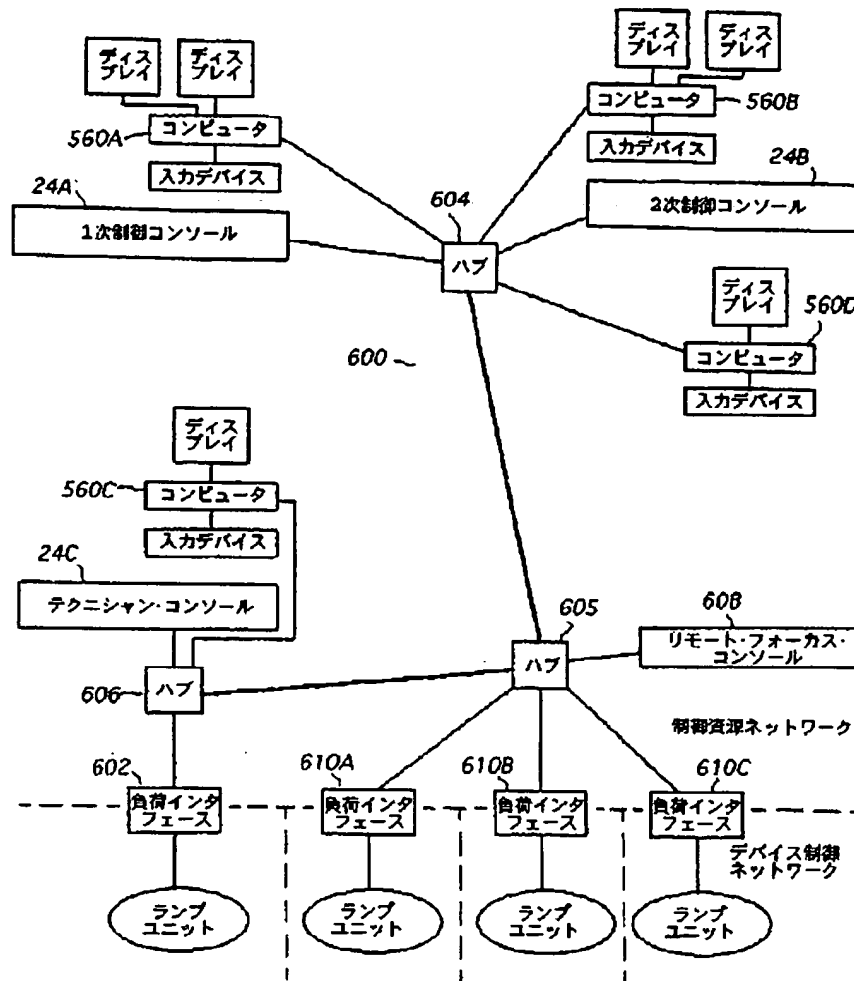
【図26】



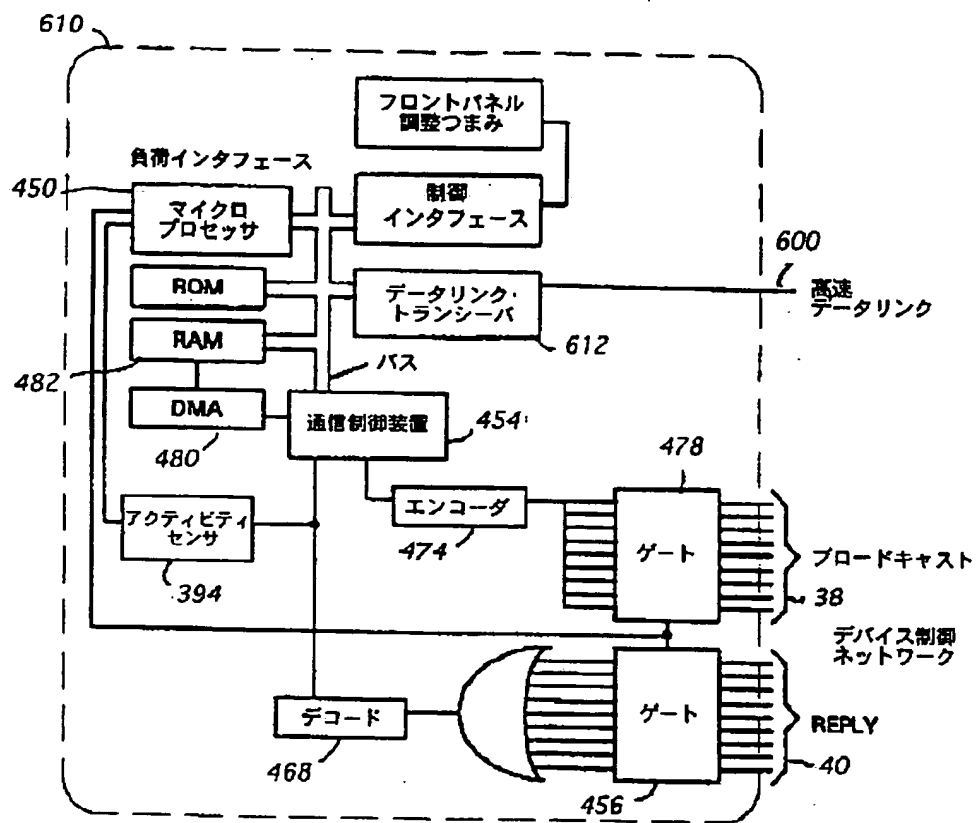
【図27】

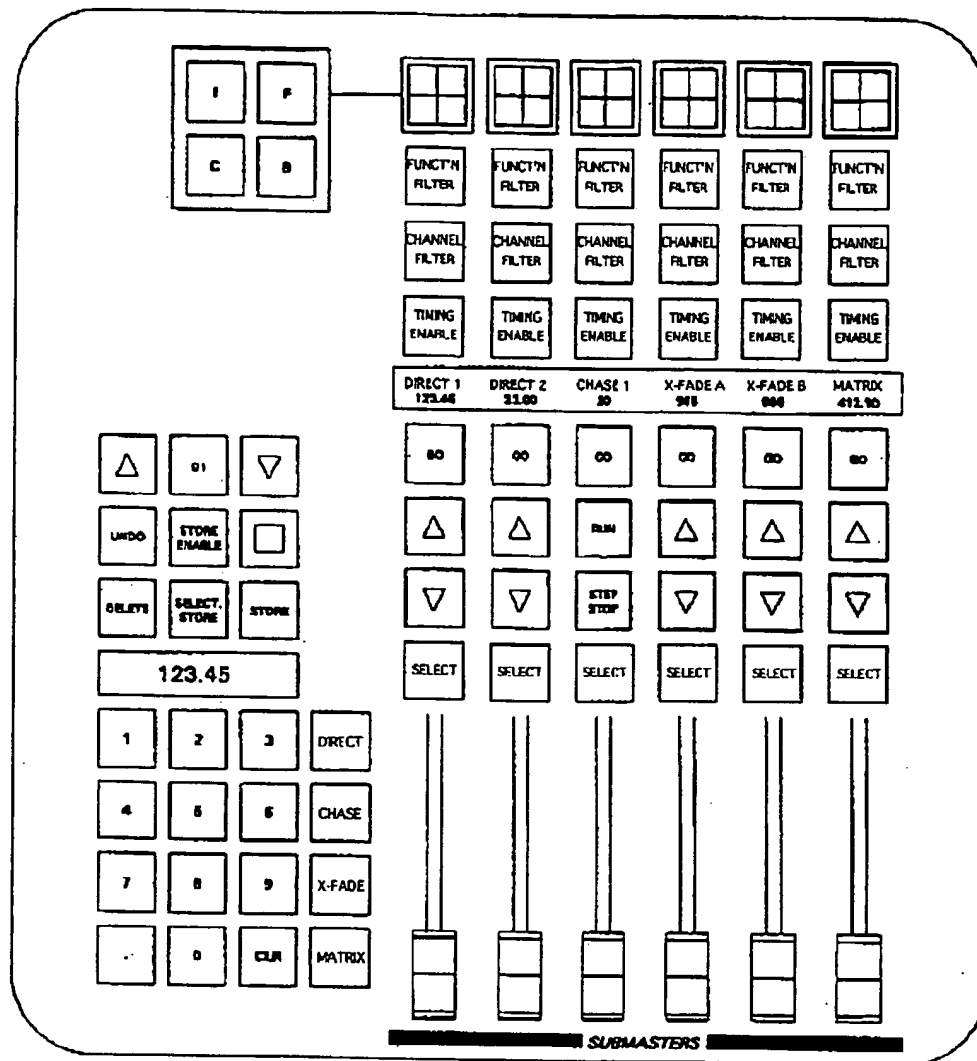


【図28】

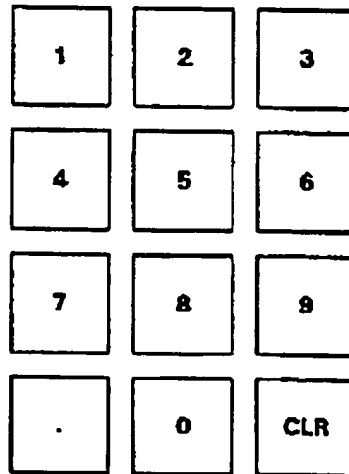
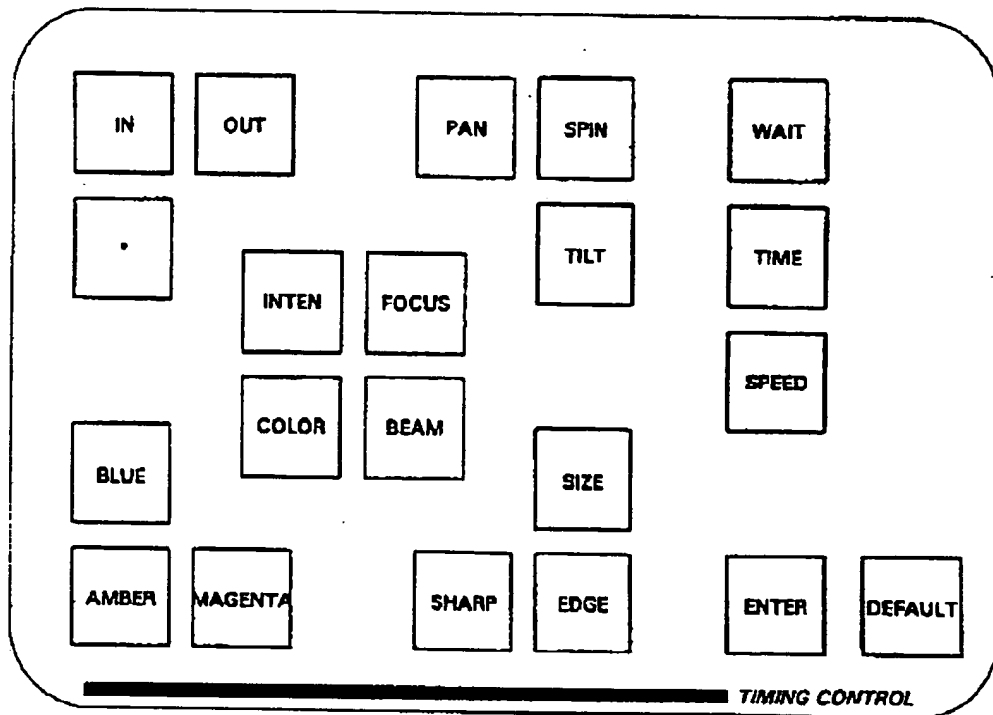


【図32】





【図 3 4】



フロントページの続き

(72)発明者 マイケル ディー. ネルソン
アメリカ合衆国 75218 テキサス州 ダ
ラス イーストン ロード 305

(72)発明者 チャールス エイチ. リース
アメリカ合衆国 76051 テキサス州 グ
レーブヴァイン リアン リッジ 2204

1. Title of the Invention

LIGHTING SYSTEM

2. Claims

1. A distributed control system for a lighting system, comprising:
 - A. one or more control devices for entering parameter-controlling inputs according to a specified format, said parameter-controlling inputs directing the operation of said lighting system, said control devices including a data processor coupled to said parameter-controlling inputs and a memory coupled to said processor;
 - B. one or more computing devices for storing, editing, and displaying data related to said parameter-controlling inputs, said computing devices comprising at least a data processor, a memory coupled to said processor, and a data display device coupled to said processor;
 - C. one or more load interface modules each including a data processor for controlling said respective interface module and for monitoring data link signals, each of said load interface modules supporting at least one device-control data link network;
 - D. a control-resources data link network connecting said control devices, said computing devices, and said load interface modules;

E. at least one said device-control data link network having a common path for connecting said load interface module to a plurality of multiple-parameter lamp units each having means for producing a light beam having a plurality of adjustable parameters relating to beam characteristics and drive means for controlling a plurality of said parameters in response to said parameter-controlling inputs.

2 . A method for operating a distributed control lighting system equipped with an intelligent lighting assistant, said lighting system comprising an input device, a reasoning engine, a data repository, a load interface module, a multiple parameter lamp unit and a network, the method comprising the steps of:

injecting an operator parameter command into the reasoning engine via the input device;

evaluating said operator command in the reasoning engine;

comparing said operator command to a three dimensional model resident in the data repository, the model comprising a representation of the system and a system operating environment;

calculating a system parameter adjustment based on said operator command and said three dimensional model;

composing a system command to achieve the parameter adjustment ordered by the operator command; and

transmitting said system command to the multiple parameter lamp via the load interface module and the network.

3 . . The method of claim 2 , further comprising the step of:

displaying a representation of the three dimensional model as altered by the system parameter adjustment on a graphic display device.

4 . The method of claim 2 , further comprising the step of:

implementing the system command upon receipt of an additional operator command.

5 . A method for programming operations of a distributed control lighting system equipped with an intelligent lighting assistant, said assistant comprising an operator command input device, a reasoning engine, a data repository, and a graphic display device, the method comprising the steps of:

injecting an operator command into the reasoning engine via the input device;

evaluating said operator command in the reasoning engine;

comparing said operator command to a three dimensional model resident in the data repository, the model comprising a representation of the system and a system operating environment;

calculating a system parameter adjustment based on said operator command and said three dimensional model;

composing a system command to achieve the parameter adjustment ordered by the operator command; and

displaying a representation of the three dimensional model as altered by the system parameter adjustment on the graphic display device.

6. In a distributed control lighting system connected by a network and comprising an operator command input device, a general purpose computer having a system data repository, a load interface module, and a multiple parameter lamp unit; an intelligent lighting assistant, said assistant comprising:

a reasoning engine interactively connected to said operator command input device and said system data repository, said reasoning engine configured to calculate system parameter adjustments and system commands responsive to said operator commands, and further connected to said interface module for transmitting said system commands via the network to the multiple parameter lamp unit;

said data repository maintaining a three dimensional model comprising a representation of the system and a system operating environment; and

a graphic display device, whereon said reasoning engine displays the results of implementing its calculations of system parameter adjustments and system commands responsive to said operator commands on said three dimensional model.

3. Detailed Description of the Invention

This application is a continuation-in-part of patent application Serial No. 08/273,262, filed July 11, 1994. That application is a continuation-in-part of patent application Serial No. 07/898,385, filed June 9, 1992, which is a continuation of patent application Serial No. 07/766,029, filed September 26, 1991, which is a continuation-in-part of patent application Serial No. 07/555,946, filed July 19, 1990, which is a continuation of patent application Serial No. 07/249,225, filed September 22, 1988, now U.S. Patent No. 4,980,806, which is a continuation of patent application Serial No. 07/120,743, filed November 12, 1987, which is a continuation of patent application Serial No. 06/887,178, filed July 17, 1986, now abandoned.

[Technical Field to Which the Invention Belongs]

The present invention pertains in general to stage lighting systems having intelligent remote lighting fixtures and intelligent data distribution networks, and in particular, to the modular control of such remote lighting fixtures and data distribution networks.

[Background of the Invention]

High performance computer-controlled lighting systems, such as disclosed in U.S. Patent No. 4,980,806 to Taylor et al., can readily handle the tasks associated with the remote control of multiple-parameter lighting instruments, including the communication of large amounts of data for simultaneously executing multiple parameters in hundreds of lamp units.

However, there exists a demand for increased flexibility in such systems so that individual system components can be upgraded or replaced in an efficient manner, without affecting the operation of the entire system.

In addition, increased flexibility is desired in such systems so that various aspects of the lighting systems can be reconfigured, as necessary, to accommodate the varying requirements of different performances.

Controllers for modern lighting systems often must be capable of simultaneously supporting diverse lamp units having different data requirements. Increasingly, lighting designers are demanding to utilize a variety of control devices and lamp units in their lighting systems in order to achieve desired lighting effects, often including lighting equipment produced by different manufacturers, each having their own unique communications protocols and data formats.

[Problems to be Solved by the Invention]

Unfortunately, however, limited space and manpower often makes it impractical to utilize separate lighting controllers to control the different types of lighting equipment used in a

single performance. In addition, many operators of lighting consoles demand a standard interface for controlling the different types of lighting equipment used in a single performance in the same manner.

Unfortunately, the development of a standard lighting controller, capable of controlling the varying types of lighting equipment provided by different manufacturers, has been impeded by the diverse communications protocols and data parameters associated with the lamp units provided by the various manufacturers of lighting equipment.

Accordingly, it is an object of the invention to provide a lighting controller constructed of independent, replaceable and reconfigurable modules.

A further object of the invention is to provide an interface system for coordinating communications between one or more control devices and a plurality of lamp units having diverse communications protocols, functions and data formats.

Yet another object of the invention is to provide an improved means for controlling a number of different types of lamp units.

An additional object of the invention is to provide a lighting controller that facilitates the designation of parameters for diverse lamp units.

A further object of the invention is to provide a lighting controller having a standard user interface for accessing diverse lamps units in the same way for each type of lamp unit.

In one implementation of the invention, a distributed control system for a lighting system is provided, including: one or more control devices for entering parameter-controlling inputs according to a specified format, the parameter-controlling inputs directing the operation of the lighting system, the control devices including a data processor coupled to the parameter-controlling inputs and a memory coupled to the processor; one or more computing devices for storing, editing, and displaying data related to the parameter-controlling inputs, the computing devices including at least a data processor, a memory coupled to the processor, and a data display device coupled to the processor; one or more load interface modules each including a data processor for controlling the respective interface module and for monitoring data link signals, each of the load interface modules supporting at least one device-control data link network; a control-resources data link network connecting the control devices, the computing devices, and the load interface modules; and at least one device-control data link network having a common path for connecting the load interface module to a plurality of multiple-parameter lamp units each having means for producing a light beam having a plurality of adjustable parameters relating to beam characteristics and drive means for controlling a plurality of the parameters in response to the parameter-controlling inputs.

In one implementation of the invention, a method for operating a distributed control lighting system equipped with an intelligent lighting assistant is provided in which the lighting system includes an input device, a reasoning engine, a data repository, a load interface module, a multiple parameter lamp unit and a network. The method includes the

steps of injecting an operator parameter command into the reasoning engine via the input device; comparing the operator command to a three dimensional model resident in the data repository, the model comprising a representation of the system and a system operating environment; calculating a system parameter adjustment based on the operator command and the three dimensional model; composing a system command to achieve the parameter adjustment ordered by the operator command; and transmitting the system command to the multiple parameter lamp via the load interface module and the network.

In one implementation of the invention, a controller is provided for a lighting system adapted to control a plurality of multiple parameter lamp units. The controller comprises:

- (1) a primary control system having source interface processors for receiving parameter-controlling inputs and a central processing system cooperating with the source interface processors for encoding the inputs into system control commands for exercising control over the lighting system;
- (2) a plurality of supplementary control units coupled to one or more of the source interface processors of the primary control system for entering parameter-controlling inputs; and
- (3) one or more of the source interface processors further comprising translating means for translating the parameter controlling inputs to meet requirements of the primary control system.

According to another embodiment of the invention, a lighting system is provided, comprising:

- (1) a control system having a central processing system for processing parameter-controlling inputs, the central processing system responding to the inputs and generating

system control commands for exercising control over the lighting system;

(2) a plurality of multiple parameter lamp units each having means for producing a light beam having a plurality of adjustable parameters relating to beam characteristics and drive means for controlling a plurality of the parameters, the plurality of lamp units being comprised of:

(A) a first set of multiple parameter lamp units having memory for storing cues, and processors for executing cues upon receipt of the system control commands identifying cues; and

(B) a second set of multiple parameter lamp units having a controller for receiving and processing absolute parameter values; and

(3) a communication system for connecting the control system to each lamp unit, the communication system includes a first load interface processor for connecting the control system to the first set of lamps and a second load interface processor for connecting the control system to the second set of lamps, the second load interface processor includes a processor for translating the system control commands into absolute parameter values for the second set of lamp units.

According to a further embodiment of the invention, a modular controller is provided for a lighting system, comprising a central control system having a central processing system for processing parameter-controlling inputs,

the central processing system translating the inputs into system control commands for exercising control over the lighting system; a plurality of supplementary control devices for entering parameter-controlling inputs according to a specified format, the parameter-controlling inputs directing the operation of the lighting system; at least one source interface module for connecting the central control system to one or more of the supplementary control devices, the source interface module supporting an independent data network that conforms transmissions to each of the connected supplementary control devices according to the specified input format associated with the supplementary control device; a plurality of multiple-parameter lamp units each having means for producing a light beam having a plurality of adjustable parameters relating to beam characteristics and drive means for controlling a plurality of the parameters in response to the parameter-controlling commands; and at least one load interface module for connecting the central control system to one or more of the multiple-parameter lamp units, the load interface module supporting an independent data network that conforms transmissions to each of the connected lamp units according to the specific communications protocol associated with the lamp units.

From the process point of view, the invention can be summarized as a method for controlling a lighting system having a primary control system from any one or all of a set

of supplementary control units having diverse signal formats, the primary control system having a central processing system, the control method comprising the steps of generating supplementary control commands at one or more of the supplementary control units; coupling the supplementary control commands to the primary control system; translating the supplementary control commands generated by one or more of the supplementary control units into a format compatible with the primary control system; and processing the translated supplementary control commands in the central processing system into system control commands for exercising control over the lighting system.

According to another embodiment of the invention from the process point of view, the invention can be summarized as a method for controlling a lighting system having a control system and a plurality of multiple parameter lamp units, the method comprising the steps of generating parameter controlling inputs; processing the parameter-controlling inputs for directing the operation of the lighting system; generating system control commands for exercising control over the lamp units; transmitting the system control commands to a first load interface processor and a second load interface processor each connected to the control system and one or more of the multiple parameter lamp units; transmitting the system control commands from the first load interface processor to the connected lamp units; translating the system control

commands in the second load interface processor into absolute parameter values; and transmitting the absolute parameter values from the second load interface processor to the connected lamp units.

Other advantages of the present invention will become apparent in the following Detailed Description taken with the accompanying Drawings.

[Embodiments of the Invention]

The present invention is an automated lighting system for providing illumination to a stage performance. Such automated lighting can provide a wide variety of illumination effects which are not possible with fixed lighting instruments. A typical installation for a computer controlled lighting system 20 in accordance with the present invention is illustrated in FIGURE 1. The system 20 is shown as it would be installed for illuminating a performance on a stage 22. The operation of the system 20 is directed by a control console 24 which serves to manually set the lighting effects of the system 20 or to automatically command the system 20 to produce a desired lighting effect determined by stored lighting cues. The console 24 is connected via a data link 26 to each lamp unit within a group of lamp units, one lamp unit being shown by the reference numeral 28.

Each of the lamp units, such as 28, have a unique address such that there can be individual communication between the console 24 and each of the lamp units. The data link 26 is

further connected to pedestal lamps, such as 30 and floor lamps, such as 32. The lamps 30 and 32 are fixed but the intensity of these lamps can be controlled by commands generated by the console 24. In operation, the system 20 causes the movable lamps, such as 28, to be adjusted individually for pan, tilt, color, intensity and beam size while the pedestal lamps 30 and floor lamps 32 are adjusted for intensity. With the addition of color-changer mechanisms 34, pedestal lamps 30 can also be adjusted for color. The system 20 is operated to provide a sequence of "cues" for illuminating the stage 22. Each lamp unit in the system 20 can have an individual response required for each of the cues. A complete performance may require the setting of several hundred cues to provide desired lighting effects.

The system 20 illustrated in FIGURE 1 shows a small number of lamp units, such as unit 28.

However, an actual stage performance may require several hundred of such lamp units. In fact, a large outdoor rock concert could require the use of up to 1000 lamp units. It can readily be seen that many thousands of commands must be generated for driving each parameter of each lamp for each of the cues within a performance. It is very possible to require ten of thousands of commands during a single performance.

The lighting effects provided by the system 20

must be properly synchronized with the stage performance to produce the programmed entertainment effect. Should any one of the lamps respond incorrectly or fail to respond, the visual effect may be destroyed. It is therefore vitally important that the lamps properly respond to the cues which are initiated by the console 24.

In previous automated lighting systems, it has been necessary for a control processor to generate each command required for setting each parameter for every light in the system. As noted above, this can require that the control processor generate tens of thousands of commands and that each of these commands be accurately conveyed via a data link to the lamps. Should there be any error in the data transmission, the lamp may respond erroneously and harm the visual effect. The electrical environment in the region of a performing stage includes many types of interference due to the heavy consumption of electrical power, for both audio and lighting equipment, in a very limited area. This electrical interference can interfere with the data transmission from the console to the lamps and can cause the lamps to improperly respond. The system 20 of the present invention is designed to overcome many of these problems while providing the capability for almost unlimited expansion in the number of lamps which can be utilized at any one time for a performance.

A functional block diagram of the system 20 as it is utilized to control a plurality of items of stage equipment is

shown in FIGURE 2. The control console 24 is connected to operate through the data link 26 for controlling a plurality of items of stage equipment. The data link 26 includes bidirectional paths 38 and 40. Path 38 of data link 26 provides data communications between the control console 24 and each of the lamp units and other units within the system 20. The path 40 provides data communication from each of the lamp units in the system 20 back to the control console 24.

In addition to the lamp unit 28, additional lamp units 42-50 are shown in FIGURE 2.

The data link 26 extends to cover a considerable area in the region of the stage 22. To maintain the integrity of the electrical commands that are transmitted through link 26, there are provided a group of repeaters 52, 54, 56 and 58. The repeaters 52-58, which are further described in detail below, provide amplification and isolation for the data transmitted through the data link 26.

The control console 24 serves not only to control the automatic lamps, such as 28, but can also be used to control a plurality of conventional dimmers such as set 60. The data link 26 is connected to a control signal converter 62 which transforms the digital signals received through the link 26 into analog control signals for directing the operation of the dimmers within set 60.

The control console 24 can also be used to control a plurality of conventional color changer mechanisms 34, such as

gel scrollers, affixed to conventional lamps 30. The data link 26 is connected to a control signal converter 63 which, like converter 62, transforms the digital signals received through the link 26 into analog control signals for directing the operation of color changers 34. Converter 63, however, is programmed to store intensity and color parameters for each control channel, and is further programmed to produce at least two analog control voltage outputs for each logical control channel, one such output being applied to one of the conventional dimmers 60 and another output being applied to a corresponding one of color changers 34.

This arrangement simplifies programming of the lighting system, since an operator can specify intensity and color parameters of a suitably equipped lamp unit by selecting a single control channel. Also, by logically separating color controlling outputs from intensity controlling outputs of the control signal converter 63, the converter can be programmed to maintain the position of the color changer mechanism while fading-out the intensity of the conventional lamp units. This eliminates the annoying effect of colors changing while fading-out the system with the Grand Master fader.

The repeaters 52-58 serve to expand the connections to the data link 26. This is termed "fan out".

Other stage action effects may additionally be controlled by the console 24. For example, the data link 26 can be connected to a control signal converter 64 from the repeater

56. Converter 64 can produce control signals for directing the operation of a chain hoist motor 66, an air cannon 68 and a special effects projector 70.

The control console 24 serves as an interface to the collection of stage devices which are subject to control. These stage devices and the associated control are termed a "Device Control Network". The control function is provided by a plurality of units which include the console 24. This group of control units is termed the "Control Resources Network". This network includes a bidirectional bus 80 which provides data communication between the control console 24 as well as additional or alternate control consoles such as 82 and 84. The direction of the system 20 can further be effected at a remote location by operation through a remote control unit 84 which is also connected to the bidirectional bus 80.

A front panel 84 for the control console 24 is illustrated in FIGURE 3. The panel 84 serves to directly control each of the automated lamps, such as lamp unit 28, or to provide automatic control for all of the lamp units. The panel 84 includes a group of key switches 86 which provides direct assignment of cue numbers for particular lighting setups. A group of rotary controls 88, 90, 92 and 94 provide color selection for a particular lamp unit or group of lamp units. Rotary controls 96, 98, 100 and 102 provide respective control of pan, tilt, intensity and zoom for each of the lamps. A group of key switches 104 provide function of preset

color selection. A particular lighting cue is entered into a console memory by operation of a store switch 106.

A grand master fade control 112 provides overall fading effects for all of the system 20 lights at one time. A black-out switch 114 turns off all lamps at one time. Cross faders 116 and 118 provide relative intensity control during a transfer from one cue to the next. The numbers of the cues involved in such a transfer are shown by indicators 120 and 122. Cue numbers are entered at the console 24 through a key pad 124. An "S" key is provided for storing a cue while an "E" key is provided for entry of a new cue. The current cue, which has been entered at the key pad, is shown by an indicator 126. A group of key switches 128 provide for the entry of cue numbers for a first cue. A group of key switches 130 provide entry of a cue number for a second cue.

The control panel 84 can take many forms provided that it allows for direct manual control of the lamp units as well as for storing and recalling of cues for the system 20.

An electrical block diagram for the console 24 is illustrated in FIGURE 4. The overall control of the console 24 is carried out by a central processing unit (CPU) 140. A representative microprocessor for use as the CPU 140 is a model 68000 manufactured by Motorola. The CPU 140 is connected to a data bus 142 and an address bus 144. The control console 24 is provided with random access memory (RAM) 146 and electronically programmable read only memory (EPROM)

148. Both of the memories 146 and 148 are connected to the data bus 142 and the address bus 144. The CPU 140, as well as other elements of the console 24, can both write to and read from the memories 146 and 148.

A hard disk drive 150 is provided in the console 24 for bulk storage of programs and data. There is further provided a floppy disk drive 152 for reading and writing conventional floppy diskettes. A controller 154 is connected to operate the hard disk drive 150 and is connected to the remainder of the circuit of console 24 through the data bus 142 and address bus 144. Likewise a floppy disk drive controller 156 is connected to operate the floppy disk drive 152 and is further connected to the data bus 142 and the address bus 144. The console panel 84, that is, the switches, lights, optical encoders, potentiometers and alpha-numeric displays thereon, is accessed through a console panel interface circuitry 158 which is connected both to the console panel 84 and to the data bus 142 and address bus 144.

Communication with the automated lamp units is carried out by use of a direct memory access circuit 164, a communications controller 166 and a Manchester encoder 168. The data bus 142 and address bus 144 are both connected to the direct memory access circuit 164 and the communications controller 166. Communication is also provided between circuit 164 and controller 166. The Manchester encoder 164 bi-directionally communicates with the communications controller

166 and also transmits and receives data to and from the data link 26.

FIGURES 5, 6, 7 and 8 are block diagrams illustrating the electronics in a lamp unit of the type which may embody the present invention. FIGURE 5 specifically shows the lamp processor, memory and associated components. FIGURES 6, 7 and 8 are block diagrams showing circuitry that specifically drives particular parameters of the light beam in a light unit.

Referring now to FIGURE 5, there is shown a lamp processor system 178. The data link 26 has the transmit and receive lines thereof connected through respective amplifiers 180 and 182. The transmit and receive lines of the data link 26 are connected through a switch 184 which is operated by a solenoid 186 that is driven by an amplifier 188. The switch 184 provides a "loop back" capability for making a direct connection between the transmit and receive lines in the data link 26 such that the lamp unit processor can perform self-testing without the use of the data link 26. The transmit and receive lines of the data link 26 are input into an encoder/decoder 190. (Harris Semiconductor Products Division Model HD-6409). The encoder/decoder 190 is connected to a lamp unit address bus 192 and a lamp unit data bus 194.

The lamp processor system 178 includes a microprocessor 200 which directs the overall functioning within the lamp unit and specifically generates the commands

which drive the mechanisms for controlling the parameters of the light unit. Microprocessor 200 is preferably a Motorola Model 68000. These parameters include pan, tilt, intensity, color and beam size. The microprocessor 200 is connected to the address bus 192 and the data bus 194. The lamp processor system 178 further includes a RAM and EPROM memory 202. The programs for driving the various parameters to the desired states and the cues for determining what these states shall be are stored in this memory. The microprocessor 200 is further connected to receive interrupts and send acknowledgements through an interrupt encoder/acknowledge circuit 204 (Motorola Model 68230) by use of an acknowledge bus 206 and an interrupt bus 208.

Interface and timing of the various circuit elements within the lamp processor system 178 is provided by an interface and timing circuit 210 (Advanced Monolithics Model 9513). The identity of a particular lamp unit is determined by a thumb wheel setting which is included in a lamp unit identity circuit 212. This identity is input to the interface and timer circuit 210. A bulb power supply 214 has various interrupt and acknowledge states which are also transmitted to the interface and timer circuit 210. The microprocessor 200 generates a series of control signals which are transmitted through a bus 216 to a decoder 218. The output from the decoder 218 comprises a group of control signals which are directed to a decoder 220 and further distributed as

control commands throughout many of the circuits in the lamp processor system 178. A group of control signals are produced by the decoder 220 and transferred as control signals to the specific control circuits shown in FIGURES 6, 7 and 8.

The data transmitted through the data bus 194 is provided to a buffer 228 which in turn transfers the data to the various parameters control circuits shown in FIGURES 6, 7 and 8.

The interrupt and acknowledge signals on lines 206, 208 are provided to a vector generator 230 which generates corresponding vector states that are transmitted through a bus 232 for transmission through data lines to the parameter control circuits shown in FIGURES 6, 7 and 8.

The interrupt signals produced on line 208 are further provided as interrupt signals to the parameter control circuits in FIGURES 6, 7 and 8. Likewise, the acknowledge signals produced by the parameter control circuits in FIGURES 6, 7 and 8 are transmitted through bus 206 to the interrupt encoder/acknowledge circuit 204.

The data bus 194 is further connected to a buffer 238 which transmits the data to both a direct memory access circuit 240 (Motorola Model 68440) and to the input of a buffer 242. The output of the buffer 242 is provided to the address bus 192. Handshake control signals are passed between the DMA controller 240 and the multiprotocol controller 246 to

synchronize the high speed communication of data to and from the microprocessor 200.

A control bus 244 serves as a bidirectional connection between the direct memory access circuit 240 and a multi-protocol communication controller 246. (Rockwell International Corp. Model 68561). The encoder/decoder 190 provides received data and received clock to the controller 246. Transmit data and transmit clock are passed from the controller 246 to the encoder/decoder 190. Various control signals are exchanged between the controller 246 and the encoder/decoder 190.

In the event an interrupt generating event occurs, the multiprotocol controller 246 asserts an interrupt output directed to the microprocessor 200. In response to an interrupt acknowledgement from the microprocessor 200, the controller 246 places interrupt vectors on the data bus 194. In a conventional manner, the microprocessor 200 temporarily interrupts processing to service the interrupt.

The multiprotocol controller 246 has serial data transmit and receive inputs in addition to a parallel system data input. The multiprotocol controller 246 of the type identified is capable of DMA data transfers up to a rate of 2 megabits per second. The high speed data stream of this nature permits the downloading of the substantial light unit cue information in a very short period of time.

The encoder/decoder 190 operates in conjunction with the communications controller 166, shown in FIGURE 4, to convert the format or protocol of the data transmitted serially through the control processor 24 into a format acceptable by the lamp processor circuit 178.

Lamp processor system 178 includes a network of clock, control and power lines (not shown) which are routinely required for the operation of a microprocessor circuit.

The lamp processor system 178 serves to initialize the entire lamp unit, command the operation of the parameter control circuits in response to manual input commands from the console or from stored cues, transfer stored cues from the memory 202 back to the control console for storage, and respond to broadcast commands received through the data link 26 for recalling cues from the memory 202 for commanding the operation of the parameter control circuits, which are shown in FIGURES 6, 7 and 8.

Referring now to FIGURE 6 there is shown a parameter drive circuit 254 which serves to operate stepper motors that are used within a lamp unit. Such a stepper motor is used, for example, for selecting color, determining iris size and selecting a gobo pattern. The microprocessor 200 has control and data paths, which are described in FIGURE 5 that are connected to a latch 256 and a timer 258. The interrupt and acknowledge lines noted in FIGURE 5 are further provided to an interrupt encoder circuit 260. The data captured by the latch

256 is transferred through a plurality of lines to a programmable array logic (PAL) 262. The PAL 262 produces a combination of control commands that are sent through a cable 264 and an enable line to a power amplifier 266. The amplifier 266 generates a series of power signals which are transmitted through a group of lines 270 to a stepper motor 272. The power signals on lines 270 cause the motor 272 to move, in a sequence of steps to a desired angular position.

The timer 258 produces timing signals required for the operation of the stepper motor. These timing signals are provided to both the interrupt encoder circuit 260 and the PAL 262. Thus, when it is required that a stepper motor change position, the microprocessor 200 produces a control command that is sent as data to the latch 254. The latched data is then transferred into the PAL 262 which converts it into control signals that are amplified by the amplifier 266 and provided to the stepper motor 272. When each operation required of the stepper motor has been carried out, an appropriate interrupt or acknowledge command is transmitted through the circuit 260 back to the microprocessor 200.

A further parameter control circuit 278 is shown in FIGURE 7. The circuit 278 is used with mechanical control parameter units which require position sensing. For the present embodiment the circuit 278 is used to control three wheels and one iris. Each of the wheels and iris has a sensor. An example is presented for operation of a wheel,

such as a color wheel, by a stepper motor. The wheel includes a mark, or magnet which is detected by a sensor 280 which is operated by an index sense circuit 282. The detected index is provided to the noninverting input of an amplifier 284. A fixed reference voltage is provided to the inverting input by operation of resistors 286 and 288. The output from the amplifier 284 is provided to a buffer 290. The output of the buffer provides address control, data, and interrupts for each of the parameter circuits to the microprocessor 200. An acknowledgement of each interrupt is provided to the buffer 290.

Referring now to FIGURE 8 there is shown a parameter control circuit 296 which provides drive and feedback control for parameters such as pan and tilt. The data bus from the microprocessor 200, as shown in FIGURE 4, provides both position and rate of change feedback and rate of change command data for a servo motor 298. The speed control data is input to a latch 300 which outputs the data to a digital to analog converter 302 which produces an analog signal that is input to the noninverting terminal of a driver amplifier 304. The driving terminals of amplifier 304 are connected to the terminals of motor 298. A tachometer 306 monitors the speed of motor 298 and provides a corresponding analog signal to the inverting input of amplifier 304. Thus, there is provided a feedback loop for determining the rate of rotation of the motor 298. The angular speed information is further

transmitted to an analog to digital converter 308 which provides the digital form of the speed information to a latch 310. The output from latch 310 is provided as a data signal back to the microprocessor 200.

The motor 298 is physically connected through a clutch 312 to a quadrature encoder 314. The two outputs from the encoder 314 are provided respectively to first inputs of amplifiers 316 and 318. The second inputs of these amplifiers are set to reference values by operation of resistors connected between the power supply and ground. The outputs from the amplifiers 316 and 318 are provided to a converter 320 which transforms the analog position signals into digital signals which are transmitted through a clock line and a direction of rotation line to an up/down counter 322. The output from the counter 322 is an indication of the position of the motor 298 and is transmitted through the data bus back to the microprocessor 200. The converter 320 further serves to produce an interrupt signal and to receive an acknowledge signal which are exchanged with the microprocessor 200.

The repeater 52, which is similar to each of the repeaters shown in FIGURE 2, is described in further detail in FIGURE 9. The purpose of the repeater 52 is to provide high speed data transmission between the lamp units, as well as other controlled stage devices, and the control console 24. The repeater 52 is connected serially with the data link 26. The repeater 52 provides bidirectional communication for the

paths 38 and 40. The lamp units and the consoles can each be considered to be both a source and a destination. The description of the repeater 52 is made in reference to the control console being a source and the lamp units being destinations.

The repeater 52 is designed to handle high speed data transmission through the paths 38 and 40 which are preferably 50 ohm transmission lines. Repeater 52 has a transmitter section 332, the upper portion shown in FIGURE 9, and a receiver section 334 which is shown in the lower portion of FIGURE 9.

The data link path 38 is connected to the input terminals of a transformer 336. Resistors 338 and 340 are connected respectively between the two conductors of path 38 and ground. Further, the data link path 38 is provided with a shield which is also grounded. The secondary of transformer 336 is connected to the noninverting input of an amplifier 342. The inverting input is connected between biasing resistors 342 and 344. A capacitor 346 is further connected between the inverting input of amplifier 342 and ground.

The output of amplifier 342 is connected to the input of a Manchester encoder circuit 352. The output from the Manchester encoder circuit 352 is passed through an inverter 354 to one or more differential current line drivers. The output of inverter 354 is connected to one such line driver 356. The output from the line driver 356 is further connected

into the path 38 for transmission to another repeater, such as 52, or to an ultimate destination such as a lamp unit.

In the receiver section 334, the path 40 is connected to the primary terminals of a transformer 358. Resistors 360 and 362 are connected between the conductor lines of path 40 to ground. Again, the shield of link 40 is grounded. The secondary of transformer 358 is connected to one input of an amplifier 364. The second input of amplifier 364 is connected to the junction of resistors 366 and 368. A capacitor 370 is connected between the junction of resistors 366 and 368 and ground.

The output signal from the amplifier 364 is passed through an inverter 372 to the input of a Manchester encoder 374. The output from encoder 374 is further passed through an inverter 376 to the input of a differential line driver 378. The outputs from line driver 378 are connected to drive the differential terminals of path 40 of data link 26. The path 40 is directed to the control console 24 or to the receiver section of a further repeater, such as repeater 52. The Manchester encoders 352 and 374 are driven by an oscillator 382 which provides inputs at a clock rate of 16 MHz. The repeater 52 further includes a startup circuit which comprises a series combination of a resistor 384 and a capacitor 386. These series components are connected between the positive voltage supply and ground. An inverter 388 has the input thereof connected to the junction of resistor 384 and

capacitor 386. The output of inverter 388 is connected to the CTS inputs of encoders 352 and 374. The output of inverter 388 is further connected to the input of an inverter 390 which has the output thereof connected to the reset inputs of the encoders 352 and 374. At power-up, the reset signals to the encoders 352 and 374 are at an initial low logic level for a short period of time. When the capacitor 386 is charged, the reset logic state changes and goes to a high logic level for normal operation. Thus, the digital circuits the Manchester encoders are set to predefined states when power is initially applied.

In a selected embodiment of the present invention, the Manchester encoder/decoders, such as 352 and 374 as well as encoder 168 shown in FIGURE 4, comprise an integrated circuit model HD-6409 manufactured by Harris Semiconductors Products Division. The Manchester encoders 352 and 374 have the mode select input connected to a logic high level thereby selecting the repeater mode. The Manchester circuit operates by receiving the high speed data stream for conversion into the nonreturn to zero (NRZ) form. The clock signal is recovered from the data stream in a conventional manner. The data stream is then retimed and reconstructed before being output to the inverter. In this manner any distortion in the nature of pulse width, delay or otherwise is not compounded through the transmission in the data link. The reconstruction and retiming of the high speed data stream at each repeater serves

to significantly reduce the data error rate through the data link 26.

In accordance with a primary feature of an embodiment of the invention, there is provided a decentralized control over the operation of each lamp unit. By this it is meant that high level commands are dispatched by the console processor to the lamp units. This is termed a "broadcast command." Each lamp processor responds in an appropriate manner defined by the program and previous condition at that particular lamp processor. This is in contrast with prior art systems wherein the console processor stores all of the current information and data concerning the status of each lamp unit and each parameter within each lamp unit. In these prior art systems, all the cue storage of data information has been handled completely by the console processor itself, and the only data that was transmitted to the pertinent lamp units were the very detailed instructions, such as the number of pulses necessary to rotate a particular stepper motor a desired number of degrees. This is to be distinguished from the system according to an embodiment of the present invention which is configured such that the console reads its control inputs, and upon sensing a change does minimal processing of the changed input (such as providing the ordinal number of a switch or the identifier of a fader) and transmits this change signal to all lamps units simultaneously, in a single high level message. Each lamp unit then recognizes the intended

effect of this change and calculates the desired response within its own processor. In processing a high level command, each lamp unit processor requires no interaction with the other lamp units, or with the console. For example, a single message that a fader on a console has been moved is transmitted to all the lamp units simultaneously. Each lamp unit processor recalculates the balance of the recalled cue information based on the individual involvement with the cue. Various lamp units may have different actions for one cue, some lamp units may not be active at all. With this new configuration, all cue memory for instantaneous recall is maintained in each individual lamp unit memory. Each lamp unit thus has available all cue information within the unit itself. However, for backup and long-term or secondary storage, the console processor maintains a copy of the cue data for each lamp unit. This backup is maintained on a disk storage and is read into the memories of the lamp unit upon system initialization at lamp replacement or for a complete memory change over.

It can be seen from the foregoing that the efficiency and reliability of the system has improved since the large body of cue data is transmitted through the narrow band-width communications link only once, namely, at system initialization. Thereafter, the cue data is available within each lamp unit, where the reading and writing thereof is performed in the environment of the high band-width local

memory. It is seen from the foregoing that the efficiency of the system is optimized, especially in situations where there is a concurrency of activity of each lamp unit in response to a newly generated command. The command from the console is simply transmitted to each of the lamp units in a system-wide manner as a broadcast command in one transmission. The activity required in each lamp unit is carried out independently of the activity in other lamp units, and without further data transmissions from the console. This results in a considerable saving of time and enhancement of reliability. This is due to the parallelism in the data link transmission. Moreover, the addition of more lamp units to the system does not significantly burden the console processor nor the data link. The system is always maintained in an optimum manner upon the addition of lamp units since each lamp unit adds the necessary processing power and memory required for carrying out its function. Very little additional load is added to the work of the console processor when a lamp unit is added to the system.

With the foregoing in mind, the console will be described next in connection with the functions of the processor system. FIGURE 10 depicts the primary functions of the console processor complex in flow chart form. On initial power up of the console, the console circuitry is initialized with predetermined internal variables, whereupon the processor enters the main sequencer program. This program is in the

nature of an endless loop which branches out to other subsidiary programs in a predefined and unvarying sequence. When each subsidiary program is entered in the sequence, it performs a specific function before returning to the main sequencer loop.

One of the subsidiary programs of the console is the switch input sensing program. This program performs a complete scan of all the console switches appearing on the front panel thereof. The depression or release of any switch is sensed by the processor complex, whereupon the appropriate response routine is activated for each switch in which a depression or release was sensed. The status of each newly activated switch is transferred to the response routine.

The switch input sensing response routines are individual scripts which specify actions to be taken when a certain switch is pressed or released. Some switches are functionally grouped together and therefore employ the same response routine. In this event, the number of the switch within the common group is identified during the response routine, in which the number is used as a switch identifier in the script which is common to all switches in the group. Examples will be described below.

A second subsidiary program which the console processor enters in the sequence, is the optical encoder input scanning program. As noted above, the rotary location of various console devices are determined, and acted upon accordingly.

The rotary input devices on the console front panel comprise optical encoder/hardware counter circuits of conventional design. The optical encoder input scanning program is operable to read the counter values for each encoder, and compare the new value to the value which was stored in accordance with the previous scan. If the comparison indicates a change in the position of the rotary device, the identifier for that encoder is combined with the amount by which the value has changed, the result being sent as a command message via the network to all lamp units. The lamp units individually determine whether the change in the rotary status of the console device requires a response in the particular lamp unit.

A fader input scanning subsidiary routine appears as the third routine encountered by the main sequencer. This routine responds to the change of position of the slider fader control devices appearing on the console panel. The faders are essentially resistive potentiometers, and the sensing of the linear motion thereof is accomplished by analog to digital converters. In this manner, when the fader position is changed, a new digital encoded number will be provided at the output of the sensing circuit. It is understood that other sensing circuits can be used with equal effectiveness. The fader input scanning program reads the current input value of each fader sensor circuit, and responds only if the value has changed from the value previously stored. As with the optical

encoder input scanning program, if the sensing of the fader shows a new position, the fader identifier is combined with the actual value read from the fader, and the information is sent via the network to all lamp units as part of a command message. The lamp units each determine the applicability of the new fader value based upon the fader identifier and the lamp unit's internal state.

A pending message manager subsidiary program comprises an additional program entered in the sequence by the main sequencer. In certain circumstances, the console switches can be activated by the operator faster than the corresponding messages can be transmitted in accordance with their respective response routines. Therefore, if a response routine finds that a previous message has not been transmitted to the network by the console processor complex, a pending message packet will be generated by the respective response routine. This packet is sent when the previous message has been completed and transmitted. The pending message manager subsidiary program scans the various subsidiary programs for the existence of any pending message packets, and also scans if associated previous messages have been transmitted. A command message corresponding to a pending message packet is then dispatched by the pending message manager, when a scan finds that a previous message has been completed.

A character display control subsidiary program is entered by the main sequencer for servicing alpha- numeric

display devices on the console front panel. Several of the switch input response routines control the displays. The character display control program provides a common control interface for the response routines. In addition, the character display control program translates display data from the format used by the console system into a sequence of commands for the alpha-numeric display devices.

Lastly, there is provided a switch lamp control subsidiary program. This program controls the lamps in the various switches to indicate to the operator whether the switch is in a depressed state or a released state. In this manner, and in contrast to prior console switch systems, electrical switch contacts for carrying lamp power are not required. This has a substantial effect on increasing the reliability of the many console switches. The lamp on and off data sent by the switch lamp control program is placed into the console processor complex memory by the response routines. Retrieval of the data by the switch lamp control program is also necessary for comparing with the newest scan to determine if the lamps associated with newly depressed switches should be illuminated or extinguished.

Also shown in FIGURE 10 with the subsidiary programs is a block indicating associated programs. These associated programs are enterable by various routines of the subsidiary programs. More particularly, these associated programs are entered on the occurrence of certain hardware interrupts

generated by the console electrical apparatus. Each associated program is a consolidated set of routines which provides control of various hardware functions, data structure or aspect of the console's logic state. One such associated program comprises the communications manager program. The primary function of the communications manager program is to control the transmission network between the console and the plural lamp units. The coordinated transmission of data to the network demanded by the various response routines is important to assure an orderly flow of information in accordance with the urgency of demands imposed by the respective response routines. The parallel nature of the transmission network is highly desirable insofar as a failure of one lamp unit does not affect the transmission capability of the other lamp units. This is in contrast with the "daisy-chained" or serially connected networks typically employed. As noted above, the communication path between the console and the lamp units are full duplex paths, i.e. a transmit and receive path on which independent and simultaneous data transmissions may occur. The communications manager program has control of the lamp units and the data transmitters located therein, thus can insure that only one lamp unit, at any one time, is using the network transmission path. In accordance with the communications manager program, there are provided two types of message addresses; namely, individual lamp addresses and the broadcast address. Each lamp unit of

the system is individually accessible by the console processor complex by transmitting the unique address associated with the particular lamp unit. As noted above, each lamp unit connected to the network will receive the lamp address; however, only the address transmitted will respond. On the other hand, the broadcast address includes a lamp address field with a special value to which all lamp units in the network respond. Moreover, each lamp unit responds to the broadcast address irrespective of their individual lamp addresses.

The console utilizes broadcast messages and individual lamp unit messages for two different categories of command messages. Messages to individual lamp units are used solely for maintaining cue data on the storage disk, for reporting the status of each lamp unit and for responding to lamp units newly connected to the network. All other functions of the system are carried out by the broadcast messages. Broadcast messages, for example, are transmitted to the lamp units for placing them or removing them from manual control. Manual control of the lamp units is established by broadcasting the change command message and allowing the lamp units to respond. In addition, cue information data is recalled by the console processor complex from the units by broadcasting the cue number and allowing each lamp unit to determine whether the cue is applicable. Once the entire system has been initialized, all functions needed of the lamp units during the

course of the performance are in the nature of broadcast messages. With the architecture, the performance of a show is not impaired by the failure of one lamp unit which would cause it to continually transmit data, thereby tying up one half of the duplex network directed from the units to the console. The other half of the duplex transmission line of the network, that portion extending from the console to the lamp units, thus remains operative for transmission of console information to the units. As a result, each unit can react to the change of status of the console switches, dimmers, rotary encoders, etc. The receipt by a lamp unit of a message transmitted specifically thereto, is acknowledged by a transmission from the lamp unit to the console. In the event a response is not received from the lamp unit, the communications manager will retransmit the command message. This retransmission negates the effect of any faulty transmission by the lamp unit because of noise or other problems. However, the lack of a response from the lamp unit after several retransmissions by the console processor complex is taken as an indication that the lamp unit is no longer operational. Selected messages transmitted by the lamp units will involve the transmission of data to the console. In a comparable manner, this transmission may be retransmitted by the communications manager of the lamp unit processor complex, should a simple reply by the console processor in response to the first transmission not be received. In the event of a more severe

network transmission line problem, the console transmits broadcast message at most three times to ensure the reception over a noisy communications line of at least one such message. Transmitted along with the broadcast message are sequence numbers which correspond to the number of times a message has been transmitted. The communications manager programs of the various lamp units disregard subsequent repetitious console transmissions by the use of the sequence numbers. The communications manager program within the control complex receives console message in accordance with the various console programs, and enqueues such messages for transmission to the lamp units. If a particular message requires a reply from a lamp unit, the console processor will wait for the reply and, when received, pass it back to the program initiating the message before transmitting subsequent messages.

Shown in FIGURE 11, which illustrates the associated programs, is a file manager program. The file manager program oversees the disk file system, and provides sequential, relative record and key indexed files for the lamp unit cue data. The cue data associated with each unit lamp is identified by a file identifier which includes the console control channel to which the unit is assigned. Programmed console data is also stored on the disk by files, one for each programmable console function. In all other respects, the file manager operates in a conventional manner. The

associated programs in the figure also include a disk data manager program. In a conventional manner, the disk data manager provides the functions of managing the list of free sectors in the disk, allocating the sectors to various files, and locating a desired sector of a file and issuing the disk hardware signals necessary to execute appropriate actions. This program requires modification to control the different disk drives employed in various implementations of the invention. Another associated program is shown in the figure as the exception display manager program. The exception display program usurps command of one of the alpha-numeric display devices located on the front panel of the console for drawing attention thereto of the operator. These situations generally arise during operation of the console where the operators acknowledgement or assistance is required to resolve a problem. A script of display data for display on the alpha-numeric devices is provided to assist the operator. The displayed data may include expected switch input responses which require activation. Once the problem has been resolved, control of the alpha-numeric display device is returned to the character display program.

The network state control program maintains management over the connection or disconnection of lamp units to the network. When a lamp unit connection is first detected by the communications manager program, the network state control program is signaled, in which event a sequence of checks is

instituted on various status bits reported from the newly connected lamp unit. These bits represent certain conditions and actions which are prerequisites of the console to recognize a fully operational lamp unit. Response routines are provided for each of these status bits. The response routines specify actions for the console to take, based upon appearance of the respective bits. Examples of some of the functions performed by the network state control program are the downloading of additional lamp unit program code, the downloading of cue data for the lamp unit and the transmission of packets of data describing the current state of various console front panel controls.

The disk state manager program monitors the insertion or removal of disks from the disk drives. A console processor interrupt is generated on the insertion or removal of disks. Because of the importance of maintaining updated cue information on the disk, it is of paramount importance to the operators of the console that notification is given of situations which prohibit copies of the lamp cue data from being updated on the disk. Notification of such malfunction is brought to the attention of the console operator through the exception display manager program. Such situations may occur when the proper combination of disks is not present in the disk drives.

In accordance with the invention, there is provided a network real-time clock program which is operative to

broadcast, on a regular basis, a real-time clock information to the lamp units. The real-time clock information comprises date and time data information. This data originates from a battery powered integrated circuit in the console circuitry, and is sent to the lamp units by way of the communications manager program. The network real-time clock program is activated by a hardware interrupt.

During the ordinary sequence of a production or show, the console regularly requests lamp status data from each lamp located on the console. Certain status bits, such as the cue-data-download request bit, cause activation of the network state control program. Other bits, such as the bulb failure bit, result in operator notification by way of the exception display manager as noted above. Still other bits are simply stored for later examination by the console operator. The lamp status scanning program is also activated by a hardware interrupt. In response to an interrupt, the status of a lamp is requested, and retrieved. Since the hardware timer producing the interrupts operates continuously, the console processor complex has available the most recent status information from all lamp units connected to the communications network.

The operations of the multiple controller network can be illustrated by referring again to FIG. 2 and FIG. 11. The bidirectional bus 80 provides data communication between and among the control console 24 and an alternate control console

84, another control console 82 and remote control units 84. In one implementation, bus 80 is electrically configured in the same way as the data link 26, and the control console 24 is provided with a communications manager program as described in the associated programs of FIG. 11. This program serves the same function of controlling activity over bus 80 that the communications manager serves in controlling activity over the data link 26.

Two types of message addresses are provided, individual console addresses and a system address, giving the same functionality as described in the descriptions for Fig. 11, e.g., individual console is individually accessible by the main console processor complex by transmitting the unique address associated with the particular console unit. In the system address command, all consoles connected to the network can respond.

Messages sent by the control console 24 to the system address contain information including the status data messages received from the lamp units, the state of the controls on the main control console and system status data processed and formatted by the main control console. These messages allow the additional and alternate consoles and remote control unit to produce the same displays as the main control console or to display different information.

Messages sent from the additional and alternate consoles and remote control unit to the main console are of two types.

One type of message is of the same format as the messages sent by the main control console to the lamp units. These messages contain data identifying the console which originated the message. As previously described, some messages to the lamp units produce a response from the lamp unit. This response also contains the data identifying the originating console; this data permits the main control console to route the response to that originating console.

The second type of message sent from the additional and alternate consoles and remote control unit to the main console is a message to the main console itself. Some of these messages duplicate an action or sequence of actions performed by an operator manipulating the front-panel controls of the main console. These messages result in the main console sending to the lamp units the same messages that would have been sent had the controls physically been manipulated. Other messages cause the main console to modify the cue data and programmed console data which are stored in the lamp units and in the memory and on the disks of the main console.

An example of remote control unit 84 is a hand-held device which the lighting designer carries onto the stage to use for fine adjustments to the azimuth and elevation of the lamp units, ensuring that the light beam does (or does not) fall on a certain set piece or area of the stage. Another control console 82 could be a director's console, used by the lighting director during rehearsals to display data for cues

other than the one currently being performed by the lamp units or to recall cues in the lamp units when the operator is away from the control console 24.

Another control 84 is a controller as disclosed in U.S. patent application serial no. 641,031, entitled "Creating and Controlling Lighting Designs". This controller provides controls which includes prerecorded commands and hands-free execution of a performance. The disclosure in application serial no. 641,031 is incorporated by reference herein.

Another controller 84 is a control device as disclosed in U.S. patent application serial no. 693,366, entitled "Improvements In High Intensity Light Projectors". Provisions from this controller includes commands in video format. The disclosure of application serial no. 693,366 is also incorporated by reference herein.

The alternate control console 84 could be located at a position which gives a more appropriate view of the stage for certain types of performances. The provision of this alternate console would prevent the necessity of moving the main control console 24 and its connection to data link 26.

Another member of the control resources network in communication with bus 80 and its connected controllers is a unit having storage and playback facilities to store, for example, the state of the settings of the master console 24 and to recall or "play" those settings or modifications thereof, during certain modes of operation.

Other implementations of bidirectional bus 80 are possible, including a Local Area Network and a point-to-point data link between the control console 24 and a single alternate control console. Additionally, the additional or alternative control consoles or remote control unit could be implemented on a general-purpose computer, rather than the purpose-built console.

The foregoing is illustrative of the various programs available to the console processor. The following is an example of the execution of various above-described programs in response to the depression of a certain "channel select" console button by the operator. The depression of this button is operative to bring a certain lamp unit under manual control, whereupon the rotation of yet another console knob is effective to rotate the lamp about one of its axes. In the following illustration it should be realized the effect of the decentralized control of removing console functions into the lamp units. Also it will be seen that with the provision of the present invention, there is a significant reduction in the processing required of the console, compared to conventional processor controlled light systems. The sharing of tasks between the console and lamp units also results in an increase in the speed to change a system parameter. In addition, in the disclosed embodiment, the console is no longer required to sequentially process a large amount of data for every lamp unit in the system. Instead, each lamp unit processor

accomplishes the action required to achieve a change for that unit. Moreover, with the present system, the entire system can be changed in the time required by a single lamp unit. Also, because of the simultaneous transmission of messages to all lamp units, lamp units added to the system do not result in proportionately slower rate of transmission as was typical with prior systems. In accordance with the example for changing the position of a stage lamp under manual control, it is assumed that the console has performed the usual initialization routines. It is also assumed that the console processor has established communications with the lamp units, and has provided each lamp unit with all the data required for the respective initializations, and the system is operating in the endless loop of the main sequencer. In this loop, the main sequencer awaits input from the operator by way of the console devices. During its sequencing routine, the main sequencer calls the switch input sensing program which scans the switch input hardware of the console to produce a map of the switches appearing on the console front panel. In this map, set bits represent push buttons being pressed, and clear bits represent push buttons which are not depressed. This map is compared to a copy of a similar map in the memory which include the status of the switches as read on the previous scan. In comparing the present and previous maps, a third map is produced which indicates switches which have changed states between the generation of the first and second maps. If no

changes are found, the program is returned to the main sequencer. Assuming that a change has occurred, the program scans the third map, bit by bit, to identify the changed switch and to activate the corresponding response routine. The identifier and new state of the newly activated switch is passed to the associated response routine. The newly operated switch is identified as a member of the "channel select" switch group, all of which are serviced by the same response routine. The switch identifier indicates the number of the switch within the "channel select" group which along with an additional group selector specifies the control channel to which the switch corresponds. All one thousand of the console control channels are each represented by a single bit in a console memory map, and indicates whether the channel is or is not selected for manual control. Because of the pressed switch, the value of the bit for its channel is inverted, thereby selecting the lamp for manual control. In the event the lamp is already under manual control, the depression of the switch would have the effect of removing the lamp of unit from manual control. Although only one bit in the noted map has been changed, the entire map is now broadcast to all lamps simultaneously. Each lamp examines the map and determines whether its control has been changed based upon the broadcast message. After transmission of this map throughout the network, no further processing is required of the console in response to the depression of the switch.

The response routine entered in response to the depression of the switch, calls the communications manager program with a command to send a broadcast type message. The broadcast message includes a pointer to the block of memory that holds the message data. The communications manager program either initiates the transmission of the data by programmable integrated circuits which implement the communications function, or in the alternative, if a communication is already in progress, the communications manager enqueues the command and memory pointer for subsequent transmission after the current message transmission is concluded. Any additional processing required by the communications process is performed as a response to console processor interrupts from the various programmable integrated circuits. No further processing is required of the communications manager program in connection with the switch activation.

When the communications manager program has accomplished the transmission of the message, or has enqueued the message for future transmission, it returns by way of the response routine program and switch input sensing program to the main sequencer. As a result, the main sequencer is entered in the endless loop at the position previously exited when the newly pressed switch was sensed. The main sequencer continues until the lamp button pressed for manual control is released. The switch input sensing program is again entered, whereupon a

comparison of the scanning maps indicates a change in the switch state. The switch is again identified, as noted above, and the associated response routine is activated.

The response routine takes no action on the release of the switch. This is in contrast to other types of switches which cause activation of the response routine upon being pressed or released. In any event, return is made from the response routine through the switch input sensing program to the main sequencer. Again, the main sequencer resumes scanning within its endless loop. Departure is taken from the loop to the optical encoder input scanning program. The rotation of the appropriate console device by the operator is effective to cause a corresponding rotation of the appropriate stage lamp. Encoder/counter circuitry, as described above, provides a numerical input to the optical encoder input scanning program. The value produced by each encoder/counter circuit changes when the encoder shaft is turned by the operator. In a similar manner to the switch input sensing program, the optical encoder input scanning program compares the value read on each scan to the value stored in connection with the previous scan. In the event a difference is found during the comparison, an appropriate lamp command is generated. The message block includes the manual change lamp command, the amount of change, and the identifier for the particular encoder. The lamp command is then dispatched to the communications manager program as a broadcast message.

All lamp units will receive the broadcast message and determine the applicability of the message to the particular lamp unit.

As noted previously, the communications manager program processes this message by immediate transmission, or by enqueueing the message for subsequent transmission when the communications channel is clear. The console program then returns to the endless loop of the main sequencer. The foregoing constitutes the participation by the console processor in effecting the change in the lamp position as specified by the operator. All additional and subsequent processing is accomplished by the individual lamp units, as required.

The next example for illustrating the principles of the invention relate to the storing of cue data information in a particular lamp unit processor memory. This function is initiated by the console operator by depressing the "store cue" switch. As with the previously described example, the main sequencer exits the endless loop, and enters the switch input sensing program. The switch input sensing program reads a new input map and compares it against the status of the system as stored in a previous map. Accordingly, the state of the "store cue" switch is found to have been pressed. The switch is then identified and the respective response routine is called.

The response routine appropriate to the "store cue" switch checks for two necessary conditions; that "store enable" switch is also currently being pressed, and that a cue number appears in the display window above the "store cue" switch. If these two conditions are met, then the console sends a store cue command broadcast message through the network to the lamp units. In addition, the cue number appearing in the window above the "store cue" button is also broadcast in the same message.

The communications manager program effects a data transmission through the network of the broadcast message, which message is received simultaneously by all lamp units. After the message is either transmitted, or enqueued for subsequent transmission, the communications manager program returns through the response routine and switch input sensing programs to the endless loop of the main sequencer. The main sequencer routinely services other operator commands as the need requires. However, in the services of this example, as well as many other that may be interspersed therebetween, the main sequencer is periodically preempted by regularly timed interrupts which require somewhat immediate attention. The regularly timed interrupts may be in the nature of the activation of the lamp status scanning program by the periodic interruption of the hardware timer integrated circuit which produces an interrupt to the console processor. On each interrupt generated by the hardware timer, the program

commands a different lamp unit to send a message to the console containing data describing the lamp unit's current status. The type of data appearing in this message is described in more detail below in connection with the lamp processor system.

Because of the store cue command broadcast, as above described, some lamp units of the system will begin reporting the occurrence of new cue data to send to the console for storage on the disk. The lamp status scanning program handles all the lamps in the system in turn, and all lamps involved in the newly stored cue will eventually be able to send their cue data to the console. The lamp status scanning program obtains the status data for an individual lamp unit by sending a status read command message to the lamp unit through the communications manager program.

The status read command message is individually addressed by the communications manager in much the same way as described above in connection with the broadcast messages. However, since the status read message command requires a response from the particular lamp unit, the communications manager program holds the communications network channel open after transmitting the lamp command message. The communications network channel is held open until the lamp replies, or until a certain time period has elapsed with no reply. In this event, a lamp failure is assumed to have

occurred. Further processing in the lamp status program is held in abeyance until a reply is received from the lamp unit.

Once the particular lamp unit has replied to the status read message, the communications manager returns to the lamp status scanning program with the message received. In this example, one of the bits in the receive message will indicate that the lamp has stored cue data in the lamp unit processor memory, which cue data has not yet been transferred to the console for disk storage. In a manner like many of the input scanning programs of the console, the lamp status program reacts only to changes in input values. The appearance of a set bit in the lamp status data causes the activation of the network state control program. A response to the change in the lamp status is thereby produced. The network state control program is provided with a group of response routines which handle the status bits received from the lamp unit. Some of these response routines provide to the console operator notice of lamp problems, such as bulb failures. Other response routines of the network state control program download program code to the lamp units, on request. The response routine associated with the data bit received in this example uploads cue data from the lamp unit, and stores the data in the proper file of the disk file system. The network state control program first checks a flag located in the console program disk state manager to insure that new cue data from the lamp unit can actually be stored. If indeed the disk

is available for cue storage, the response routine then calls the communications manager program with a cue buffer upload message, as well as a pointer to an unused section of memory in which the data is to be stored. In the event the disk is not available for storage of cue data, the new data is not uploaded from the lamp unit. Instead, the front console panel indicator is illuminated, whereupon the operator is reminded that cue memory is required to be uploaded from the lamp unit to the console. This can be accomplished later by an operator command.

The cue upload command, much like the status read message described above, is sent to the particular lamp unit. The cue upload command also causes the communications manager to wait for the lamp unit reply. In one form of the invention, the programmable communications circuits are set up to store the lamp unit reply in the memory space specified by the network state response routine. When the transfer of the data from the lamp unit to the console is completed, the communications circuitry interrupts the console processor. The communications manager program is reactivated. The communications manager program thus determines that the communication transmissions is complete, commences the transmission of another message, if such a message is pending, and returns to the network state control response routine.

By the involvement of the network state control response routine, the data received from the lamp unit is subdivided

into file records. The same format employed to subdivide the file into records is used both in the lamp cue storage as well as in the disk file system. In certain situations, the data received may be that of several cues, since the rate of scanning the lamp status can be temporarily slower than the rate at which the operator is storing cues. In this example, it is assumed that only the data from the cue store command is the data being operated upon. The disk file already contains the lamps cue data as it existed before the cue data to be stored. Therefore, all that is required is to add or rewrite the appropriate record in the disk file. The response routine accomplishes this by calling the file manager program to open the file with the particular lamp control channel number in the cue data file directory. The foregoing is accomplished by the response routine in calling the file manager program to open the file with the lamps control channel number in the cue data file directory. The response routine then issues a write command to the file manger program, using the record data received from the lamp unit. Once the writing of this data is accomplished, the response routine calls the file manager program, and the file is thereby closed.

The file manager program performs the three functions for the cue upload response routine, as described above. The command to open the cue data file results in a search for the file descriptor in the directory of cue data file descriptors. When found, the descriptor is used to find the first fragment

of the file and load it from the disk. The file record to be written comprises two parts; the cue number, and the lamp function data. The cue number is utilized as a unique index to the record. When the command is issued to write the newly received record into the file, the file manager program searches the fragment already appearing in memory to find the index of the record being written. If the index is not found in the first fragment, the other fragments of the file are examined in turn. If an existing record already contains the cue number of the record being written to the file, it is overwritten with the lamp function data of the new record. If the index is not found in the file, the record is added to the file. The command from the network state control response routine to close the file causes the file manager program to release the pointers to the data in memory relating to the file. In this manner, the network state control response routine can reuse those memory spaces whenever needed. No further access to the file can be made without issuing the file open command.

Whenever it is necessary for the file manager program to access the data stored on the disk, as opposed to the copy in the console processor memory, the disk data manager program is activated. This program provides control of the disk drive controller circuits which actually issue disk commands and reads the data from the disk. The disk data manager maintains an account of those parts of the disk currently being used,

END

and determines the actions needed to access specific file fragments requested by the file manager. Finally, the response routine is terminated, and is returned through the network state control program to the lamp status scanning program, which is also terminated until the next timer interrupt. The foregoing describes the operation of the system, assuming the initialization of the lamp units has taken place. The detailed initialization of the lamp units are described in detail below. Each lamp unit is initialized during system power up and initialization, or when added to a functioning production light system. As described above in connection with the circuitry of each lamp unit, there is provided a processor and sufficient memory for storing various programs, which, when carried out, allow any unit device to be moved, readjusted or changed in accordance with a cue or switch actuation originating at the console.

Referring now to FIGUREs 12 and 13, when power is applied to the system as a whole, or to a lamp unit, the lamp state initialization program is activated. This program may also be activated during normal lamp computer operation when certain interrupts occur indicating a major malfunction of the lamp system. In addition, part of the lamp state initialization program is re-entered if the communications address of the particular lamp unit is changed.

Each lamp unit includes a ROM-based program which performs various functions. For example, the ROM-based

program tests certain hardware necessary for the proper operation of the lamp system, the program presets various programmable circuits in the unit to predetermined known states. In addition, the program proceeds through a script calling for checks to be made on certain parts of the unit, and prescribing actions to be taken depending upon the results of the tests. At the end of the script, the lamp unit is in complete synchronization with the console, whereupon the processor enters an endless loop consisting of self tests, physical state monitoring and response to console command transmissions.

The first task performed by the initialization program is a checksum test of the validity of the programs from the EPROM memory. A test of the hardware timers against pretimed software loops is also performed. A loop back test of the communication hardware and a read/write test of part of the RAM memory is also conducted. If any of the lamp units' circuitry tested is found to be faulty, execution is halted. Once the operation of the lamp unit hardware has been tested, various program subroutines are executed to initialize program variables, and set up programmable circuits used for communications. The identity of each lamp unit is a communication address read from an appropriate input device. In one form of the invention, the identity of each lamp unit is established by the setting of a three-digit thumb wheel switch. Thus, as many as one thousand lamp units can be

connected to the system, and retain an independent identity. An input to the lamp complex processor represents the configuration of servo and stepper motors associated with the lamp hardware. Since the same computer hardware and basic programs are utilized to control different combinations of actuators of the lamp complex, a portion of the lamp system programs will differ between the lamp complexes. If needed, the proper system programs for particular lamp complexes can be downloaded from the console. However, the downloading of these additional programs into the lamp units may not be necessary, as the programs are maintained in writable, nonvolatile memory in each lamp unit. As noted above, the nonvolatility of the memory is provided by battery RAM memory backup. A check is conducted next on the programs already present in the lamp unit RAM memory to determine the validity thereof. A checksum test is performed and identifiers in the programs are matched with the actuator configuration input noted above. If the programs are found to be valid, an internal flag is cleared, thereby allowing the execution of the additional programs. If additional programs are found to be invalid, a flag is set in the memory status word and the console performs a download of the program for replacement of the lamp system program memory. The flag which disables the execution of these additional programs is also set.

At this time, a communications manager program is activated to establish contact with the console processor

complex. Thereafter, when the console interrogates the communication address of the lamp unit, the communication manager program will respond. The particular configuration of the lamp unit, and the results of the above-noted validity checks are reported in response to the console command. This constitutes one of the initial communications between the console processor complex and that of the lamp unit.

In the event additional programs in the lamp unit RAM memory are found invalid as a result of the above checks, further execution of the initialization is postponed until the programs are downloaded from the console. The lamp unit processor enters an endless loop of self tests and console command responses. At the end of the command response routine program associated with the program download, the flag which was previously set to disable execution of the additional programs in RAM memory, is cleared. The lamp state initialization script is then reentered. Eventually, a valid set of these additional programs will exist in the RAM memory of each lamp unit. Subroutines associated with the additional programs are then run to initialize additional program variables and the programmable circuits used for control of physical actuators. A table of address which guide the lamp unit processor to interrupts is also modified to reflect the presence of interrupt response routines in the additional programs. More subroutines are then called to perform calibration and indexing functions of the physical actuators

and feedback sensors. The subroutines cause the various actuators to be moved through their full range of motion, noting the location of any sensors and checking for proper operation of the various actuators and feedback sensors.

In the event the communications address for a lamp unit is changed during operation of the lamp, communications are reestablished with the console in accordance with the new address. The lamp state initialization script is reentered to allow resynchronization of the lamp unit with the console for the new address.

A flag in the lamp status word is set at this time to prompt the console processor complex to transmit a packet of data containing information concerning the state of the console. This packet of data is necessary to the lamp unit to allow it to respond properly to subsequent console commands. The nature of the data in the packet comprises information relating to the position of controls in certain subsections of the console, and the console control channel number assigned to the particular lamp unit. A flag is set in the intensity logical controller to prevent the light of the particular lamp from being turned on, until adequate data has been received from the console. The initialization program then reenters the self tests/command response loop until the receipt of the console state packet.

On completion of the console state packet, command response routine, the lamp state initialization script is

reactivated. The data associated with the state packet received from the console is stored temporarily while additional validity checks are performed on the cue data memory. A checksum test is conducted, and a test for a match between the control channel identifier in the cue data, with the control channel identifier received from the console. If the cue data is found to be valid during the checksum/channel-number test, a notation of the time of the last update to lamp unit cue data is compared with that of the data stored on disk in the console. If these update times match, processing continues. In the event that more recent data is found to be stored in the lamp unit memory, console operator arbitration is invoked to determine which cue data should be used. If it is decided that more recent data is present on the disk, than in the lamp unit memory, or if the cue data is found to be invalid, a flag is set in the lamp status word. This flag prompts the console processor to download the proper cue data into the lamp unit memory. A rewind command is then sent to the cue data manager program to erase the data in memory and the self-test/command-response loop is reentered.

In the alternative, when valid cue data found to be present in the lamp unit memory, the initialization script is reentered, whereupon the cue data and the console state packet are utilized to set up all function logical controllers to respond to the next manual control or cue recall command from

the console. When the cue recall command is received, a flag in the intensity logical controller program is cleared. As will be recalled, this flag suppresses the illumination of lights not fully synchronized. Normal operation of the lamp unit is then allowed to commence. After this final program setup, the initialization script is terminated, and processing continues within the main sequencer loop until the occurrence of one of the activation criteria, described above.

The foregoing describes in general the console and lamp unit processor interchange for accomplishing the proper initialization of the lamp units. After initialization, the primary background activity performed by each lamp unit is through each respective main sequencer loop program. Refer now to FIGURE 14. Generally, the activities of the lamp unit processor in the main sequencer loop include scanning input buffers for communications received from the console processor, the performance of checksum integrity checks on both cue data and program code in the RAM memory. Also, the lamp unit processor scans changes in the communication address associated with the unit. The main sequencer loop is the program which is continuously executed in each unit, until a console command communication is received or checksum failure or address change in which event the main sequencer loop is temporarily exited. In addition, processing within the main sequencer loop is temporarily halted when interrupt-based

actuator control programs are activated, or on the occurrence of physical feedback interrupts

. The main sequencer loop program itself is an endlessly repeating preset cycle for activation of a variety of subprograms. The subprograms are discussed in detail below, and include the command interpreter, the memory checksum test and the communications address scanning subprogram. In each case, when the main sequencer loop enters the subprogram, a test is conducted, in which event the main sequencer loop is re-entered, or a response is performed based upon the result of the test conducted.

With regard to the command interpreter subprogram, an endless loop type of program is activated, in which event a sequence of instructions are performed. The first instruction or action performed in the command interpreter subprogram is the issuance of a read command to the communications management program. After the read command, a return to the main sequencer loop is executed. On subsequent activations of the command interpreter subprogram, checks are conducted with the communications manager program on the status of the previously issued read command. Return is made to the main sequencer loop on an indication of the uncompleted processing of the read command. On an indication of the completed processing of the read command, i.e., when the check status reveals a completed communication from the console processor, the command interpreter subprogram examines the first word of

the newly received data issued by the console command message, and to be executed by the lamp unit processor. If the console command is of the type which requires no further data transmission from the console, the received data is temporarily stored, and another read command is issued to retrieve the next command sent by the console. Those console commands which are received and which have associated response routines stored in the lamp unit ROM memory are performed immediately. The validity of additional programs located in RAM memory is verified before performance of other console commands. In any event, processing continues in the command response routine until the command is complete, or until all further processing of the command is interrupt-based. In this event, control is returned to the main sequencer loop. Particular types of console commands, and their associated response routines will be described below. When activated, the memory checksum subprogram verifies the integrity of memory sections having stored therein program code and cue data. Tests are performed only on those sections of memory believed to be valid. If a checksum test of the program code fails, an appropriate flag is set in the lamp status word to prompt the console to download the program code. Furthermore, operation of the command interpreter program is limited until the program code is replaced, and thus again validated. When the console responds with the necessary download of the program code, the

lamp state initialization script is reentered, as described above. In the event that the cue data is found to be invalid, the appropriate flag is set in the lamp status word, wherein the console processor is prompted to download cue data. A rewind command is dispatched to the cue data manager program to clear the invalid cue data. No further processing is required after the download of valid cue data. In both situations, once the appropriate actions have been accomplished, control is returned to the main sequencer loop.

The subprogram identified as the communication address scanning program reads the identification code of the lamp unit. As noted above, the identification code is established by a digit switch initially set to provide a unique address for the lamp unit within the communication network. This subprogram compares the value read from the switch with a copy in the memory. If the comparison shows that the identification address has changed, a timer is started. This timer will produce an interrupt of the lamp unit processor after a certain period of time. The new identification address read during the scan is stored in the memory for comparing with subsequent identification changes. In each instance in which a new identification address has been detected, the timer is restarted. No other response is necessary when the communication address is altered, until the timer interrupt occurs. A time period of five seconds, for example, is preferable to assure that an address change has

been completed on the switch device. When the timer interrupt occurs, the lamp state initialization script is reentered. Processing of the address change occurs in accordance with the noted script, and as described above.

As noted above, the command interpreter is activated in connection with communications between the console processor and the lamp unit processor. Command response routines may activate one or more other program associated with this level of lamp unit processing. These other associated programs include the state data manager, the cue data manager, the communications manager, the function logical controllers and the physical control manager. Many of these programs report data directly to the state data manager program. The physical control manager oversees the activation of additional programs which control the physical actuators of the lamp unit, such as motors, dimmers, etc. The command response routines are individual scripts of the actions required to carry out a command issued from the console processor. This program flow is illustrated in FIGURE 15. Some routines manipulate internal data, while others transmit specified data to the console, and yet other programs perform a specific action needed to move or otherwise control the physical actuators of the lamp unit. Some of the noted routines call for a combination of the above-specified actions. In describing the following command response routines, it is important to note that a response routine is selected based on value found in

the first word of the message transmitted from the console processor to the lamp unit processor. Each of the command messages includes a unique value, known as a command identifier.

The first associated program, the state data manager routine, provides a common source and repository of status data from both the console processor and the lamp processor. Data which is received from the console processor, and which is used infrequently, is maintained accessible to the command response routine, and is retrieved upon demand. More frequently used data is passed to the function logical controllers after receipt from the console processor. Certain data, termed state data, is transmitted from the console processor in a form which includes data packed together for every lamp unit in the system. The state data is transmitted in a single simultaneous transmission to all lamp units. The state data manager extracts from the transmission, state data applicable to the particular lamp unit. The control channel assignment made by the console during the initialization script identifies the data applicable to each lamp unit. The logical and physical controllers report the various states of the unit apparatus directly to the state data manager. The state data manager combines this data from multiple sources within the unit into a single block of status data. In response to periodic console commands, each lamp unit transmits this status block to the console.

The communications manager is an associated program which has been described previously in connection with the operation of the command interpreter program. Command response routines performing the download of bulk data from the console, (RAM-based programs or cue data) issue read commands to the communications manager routine. These read commands are effective to store data sent from the console into the proper memory of the lamp unit memory. The command response routines issue write commands to the communication manager when the lamp command requires a lamp unit transmission of data back to the console. The write commands provide the proper location for access of the data within the lamp unit memory.

The communications manager routine also is responsible for the retransmission of data in the event initial transmissions were not received by the console processor. In doing so, the communications manager routine handles the fragmentation of large blocks of data, to overcome the affects of noise in communications network channels.

The cue data manager associated program comprises a conventional key-indexed file system in RAM memory. A unique, operator-assigned cue number is kept in the first four bytes of each record of the cue data file, and is used as an indexer for identifying that record. On cue recall, various indices are searched for a cue number matching that of the cue being recalled. If a match is found between the cue number

searched, and those stored, the cue data record is retrieved and returned to the command response routine. The failure to find a match between indices is likewise reported to the command response routine.

Because of the many operational features of the lamp units, there is provided a logical control program for each of the physical functions of the lamp unit. While not exclusive, the various lamp unit functions may include intensity, position, color and beam logical controllers. Depending upon the manner in which the physical hardware of each lamp unit is provided with these functions, a corresponding variety of logical control programs will be implemented. The logical control program each perform a similar function of the lamp unit, by providing a single control point for each function of the physical apparatus. The services provided by all the logical control programs include receiving cue data recalled at various front panel sources, herein referred to as submasters. The services also include the integration of new cue data with previously recalled data from other submasters, changing of the current function data according to manual control command received from the console, and reporting the current function data values. Some of the logical control programs also store current function data as preset function values, and also operate in recalling and reporting these preset values on command of the console. Some logical control programs also use fader values sent from the console processor

for proportional scaling of recalled cue data. The physical control manager associated program oversees activation of the subprograms which effect the changes in current function data, as computed by the logical controllers. The noted subprograms fall into two main categories. The subprograms controlling, for example, stepper motors, implement conventional algorithms which output a timed sequence of step commands to the motors. Some of the stepper motor subsystems will include switch closure indexing feedback for use in assuring that the stepper motors are following the stepping commands transmitted by the lamp unit processor. Other functions of the lamp unit involve the driving of dc servomotors to provide pan and tilt lamp movement. Lamp velocity information output by a dc servomotor tachometer, and position feedback information from an optical encoder/counter circuit are transmitted on the system data bus as feedback information to the lamp unit processor. The subprogram controlling these components utilizes a conventional velocity-feedback servo control algorithm. This subprogram is also activated upon an unexpected motion of the servo- controlled lamp function by way of hardware interrupts generated by changes in the position feedback signal. The unexpected motion of the servo- controlled lamp function notifies the lamp unit processor of movement occurring in the lamp, which movements were not commanded by the lamp unit processor. It is understood that these subprograms could be replaced by analog or digital circuitry.

Certain status data concerning the lamp unit will originate in the physical controllers. For example, the integrity of the bulb in the lamp unit light will be derived from the behavior of the power supply which supplies power to the light. The impeded motion of a lamp unit within its range of movement will be deduced when the motor motion fails to produce a corresponding movement of the lamp. Also, the failure of a stepper motor subsystem can be deduced from the failure of a search for an expected index input. This status information is reported directly to the state data manager.

In accordance with the two examples set forth above showing the console processor operations in response to the selection of the lamp for manual control, and for storing cue data, the two examples are repeated below for showing the actions taken by the lamp unit processors. The two examples exemplify the processing which occurs in the lamp units, and include the activation and interaction of the various programs within each lamp unit, and the distribution of tasks between the console and lamp units according to the invention.

The first lamp unit example concerns the sequence of actions occurring when the console operator selects a single lamp in the system for manual control, and the manipulation of a console device for changing the spatial orientation of the lamp unit. Both examples assume that all necessary RAM-based programs, together with the cue data, are in full synchronization with the console.

As part of the main sequencer loop, the lamp unit processor jumps to the command interpreter program to check the status of the outstanding read command in the communications manager program. The command interpreter program employs a block of memory space to service the communications manager program. This block of memory contains a byte of data which is used to signal the status of the execution of the read command. In servicing the read command, the command interpreter program checks the status byte of data in the command block being executed by the communications manager program. When a flag shows that the outstanding read command has been completed, i.e., that a block of data has been received from the console processor, the command interpreter program examines the first byte of this data. The value of the first byte of data represents the specific command to be performed by the lamp unit.

According to the example, the command received from the console is found to be a manual-controlled channel-selector-map command. Since this command does not require additional data from the console, the command interpreter program establishes another read command block, and reactivates the communication manager program. The communication manager program then prepares the lamp unit to receive another console command transmission and returns to the command interpreter. The command interpreter program then jumps to the associated command response routine. The noted

console command represents a message transmitted to the network and received simultaneously by all the lamp units connected to the network. Noteworthy, the processing described in connection with a particular lamp unit, will also be occurring concurrently in other lamp units of the system.

Because the performance lighting system of the invention can accommodate upwardly of one thousand stage lights, bytes of data must be transmitted throughout the network, one bit position being representative of each lamp unit. The location of a bit, corresponding to a particular lamp unit, is derived from the console-control-channel number assigned to the lamp unit by the console, during the lamp state initialization script. The other lamp units of the system are assigned different console- control-channel numbers, and each unit will independently extract its own bit-data from the one hundred twenty-five byte block. The console-control- channel number is stored in the state data manager program.

The action required of the command response routine, as a result of decoding the console transmission, is to jump to the state data manager program with the location in memory of the 125-byte block. Also, the command response routine provides an identifier indicating that the manual-control status bit is to be manipulated.

The state data manager is provided with a subprogram which utilizes the console-control-channel number as an index

to extract the value to be assigned to the boolean flag concerning the lamp unit selection/deselection for manual control. This boolean flag is referenced when manual-control commands are received, and either allows or disallows a reaction by the lamp unit. Control from the state data manager program is then returned to the endless loop of the main sequencer.

The lamp unit processor executing the instructions of the main sequencer program periodically enters the command interpreter program to ascertain whether a new transmission has been received from the console. It is assumed here that commands are received from the console indicating that the console operator is manipulating the lamp position controls. As a result, the next lamp unit commands received by the command interpreter are encoder-change commands. This information is determined from a first byte of the encoder-change command, whereupon a jump is made to the appropriate command response routine. Again, this command is received simultaneously by all lamp units in the network, and all such lamps will be executing the appropriate actions concurrently.

The command response routine concerning the foregoing command, first checks with the state data manager whether the boolean flag currently indicates the selection or deselection of the particular lamp unit for manual control. If the flag is not set, the command response routine terminates, and thus

the encoder-change command is ignored as the lamp is not selected for manual control. However, in the current example, processing continues as the flag is assumed to be set as part of the preceding manual-control channel-selector-map command.

The encoder-change command byte transmitted by the console in response to the change of position of a console control, is accompanied by a byte of data identifying the particular console encoder. This is essential as there are several encoders on the console panel. Each encoder provides control of a different lamp function. In addition, the encoder-change command byte contains data representative of the amount of change in the encoder input value. Because each encoder is associated with a different function of the lamp unit, the command response routine executes a jump to the function logical controller associated with the encoder that has a changed input value. The command response routine also passes along the data corresponding to the amount by which the position of the lamp unit is to be changed.

With regard to the present example, the position logical controller is activated. The position logical controller reads the data representing the current command position of the lamp unit, and modifies this data by an amount linearly proportional to the received encoder change input value. This new value is now stored as the new position of the lamp unit, whereupon the position logical controller returns to the command interpreter.

Next, the command interpreter program activates the physical control manager which compares the command data presently stored in memory with all the logical controllers with the actual positional states of the lamp unit physical devices. The actual states of the physical devices are brought into conformance with the commanded states. In those situations where more than one function data has changed, the physical control manager will activate the physical actuator programs in preprogrammed combinations in order to ensure that all such actuators perform properly.

In the present example, only the servomotor control program is activated. This program calculates the direction of change called for by the new command data, as well as the appropriate magnitude of the voltage to be applied to the servomotor. An associated timer is also triggered to provide periodic hardware interrupts. At each interrupt, the servomotor control program will recalculate the appropriate voltage to be applied to the motor, until the actual state of the servomotor subsystem matches the command data established by the position logical controller.

The foregoing institute the motion of the desired servomotor to effect a corresponding change, for example, in the pan or tilt position of the lamp unit. Once servomotor motion is initiated, the lamp unit returns from the servomotor control program and the physical control manager program to the command response routine and the command interpreter.

Control is returned from the latter two programs to the main sequencer where the scanning for received commands, memory checksum failures and communications address changes processing resumes. Until such time as the new lamp unit position is reached, the hardware interrupts and servomotor control recalculations are interspersed with the actions of the main sequencer's endless loop. The high level commands transmitted by the console through the network and to each lamp unit, the commands undergo additional processing in each unit to determine the effect of the command on the unit, and to accomplish the desired result, if applicable.

The next example involves the processing in the lamp unit as a result of the console operator having actuated the "store cue" switch on the console panel. The lamp unit processor exits the endless loop of the main sequencer and jumps to the command interpreter to check the status of an outstanding read command in the communications manager program. In this example, the command interpreter program discovers a newly received message from the console, having a store-cue opcode in the first byte of the command message. The command interpreter restarts the read command on the communications manager, and calls the store-cue command response routine. This command is received simultaneously at all lamp units in the network, and all such units execute the following sequence of actions concurrently.

In the command response routine, each logical controller is queried concerning the current commanded function data. This data is packed into ten bytes of memory storage area. Moreover, this block of data is combined with four bytes of data representing the operator assigned number for the cue. It should be understood that the cue number was received as part of the cue-store command transmission from the console. The command response routine then calls the cue data manager program, bringing with it the fourteen-byte block of data resulting from the above-noted processing. The cue data manager scans its list of record indices, i.e., cue numbers, for an index matching that of the record cue number passed by the command response routine. If a match is found, the accompanying data record is overwritten with the data record received from the command response routine. If no match in the index is found during the search, a new record is written into a blank record area in the index and data file. This data memory of the lamp unit is of the type which has not yet been transmitted to the console disk copy to update the present lamp units cue data. Sufficient room should be provided for several cue records should there exist a delay in transmitting the data records to the console disk storage. The cue data manager then returns to the command response routine.

The command response routine immediately calls the state data manager to set a flag in the lamp status word indicating that the lamp unit has cue data ready for

transmission to the console disk storage. The programs are then returned in seriatim through each other, until the lamp unit has threaded its way back to the main sequencer. Processing within the endless loop of the main sequencer then resumes.

At some point, in the processing of the cue store command, the command interpreter senses that a message has been received having a lamp-status-report opcode. A preparatory read command is reissued to the communications manager program, and the lamp-status-report command response routine is called. These lamp-status-report commands are dispatched individually to each lamp unit in the network, wherein only one unit will respond to the console at a time.

The command response routine calls the state data manager program to obtain the current value in the lamp status words. This block of memory is utilized as the message data in a write command issued to the communications manager program. This write command has no interaction with the read command just performed in preparation for the next console command transmission. Return is had through the various programs to the main sequencer, where the endless loop is resumed.

The flag set in the lamp status data, which flag was reported to the console in the previous command, prompts the console to issue a read-cue-data-change- buffer command. This command is received by the communication manager program of

the lamp unit, and is sensed by the command interpreter program. In addition, this command is addressed to a particular lamp, and only that lamp will transmit a response. A read-cue-data-change-buffer command response routine is provided for retrieving the list of new cue data from the cue data manager program. In addition, the noted command response routine sends the list as message data in a write command to the communications manager program, and calls the state data manager to clear the flag indicating data is present in the cue data change buffer. As a result, the lamp unit processor returns to the endless loop of the main sequencer, and waits further console commands.

The foregoing illustrates the lamp unit processor actions required to carry out a change in the position of a stage light, as well as the storing of cue data within the lamp unit memory. The flexibility of the system, however, is not limited to the foregoing. While an exhaustive description of each command is not necessary, and would only encumber the description of the invention, the other lamp commands used in connection with the lamp units of the invention are listed below.

The overall function of the programs to carry out the operation of the present invention have been described in detail in reference to FIGURES 10-15. A detailed code listing for a representative portion of the overall program is presented below. This is the code required for implementing

the color logic control which was described in reference to FIGURE 14. This code is written for execution on a Motorola microprocessor Model 68000. The color logic control program is quite similar to the logic control programs for intensity, position and beam diameter.

It can be seen from the foregoing that the lighting system disclosed provides accurate, efficient, and flexible control of several hundreds of automated lamp units. Provisions are included for the reporting of status data from the lamp units to the control console. This status data may include real-time display of parameter data including the present intensity, color, beam shape, and beam direction of the lamp units as well as any timing parameters associated with the present cue which has been recalled. Provisions are also included for the renewal of operating system programs in any lamp units which experience serious logical errors in their associated memory. Provisions are also included for the storage of parameter data associated with the various cues, which enables an operator to save the data used to execute a show and to load the data into a lighting system similarly configured but composed of discretely different lamp units which may be disposed in a different physical location from that at which the show was previously performed, for example on a different continent.

Referring now to FIGURE 16, a simple data repeater circuit (shown in greater detail in FIGURE 9), includes an

activity sensor 392 coupled to the broadcast network 38 and an activity sensor 394 coupled to the reply network 40. Each activity sensor drives a red LED 396 (via pulse stretcher circuits which make the flickering of the LED visible to the human eye), which LED's are mounted on the exterior of a repeater box and flash whenever there is any electrical activity on the branch of the corresponding network to which the repeater is connected.

The simple data repeater also includes a Manchester decoder/encoder 352 coupled to the broadcast network 38 and a Manchester decoder/encoder 374 coupled to the reply network 40. As described earlier, the Manchester encoder/decoder integrated circuit can be connected in a "repeater" mode in which messages received at its input are decoded and then re-encoded for further transmission. Each decoder/encoder drives a green LED 398 (via pulse stretchers), which LED's are also mounted on the exterior of a repeater box and flash whenever there is valid Manchester-encoded data passing through the repeater. In normal operation, red and green LED's will flash simultaneously. Any other condition, for example red LED flashing with no green LED flashing, indicates an error in data transmission. However, this arrangement can not detect subtle errors in the messages transmitted through the network.

As shown in FIGURE 17 and FIGURE 18, the data communications network 26 implemented in this automated stage

lighting system includes a control console 24 communicating with a plurality of automated lamp units (ALU) through several intervening repeater circuits. A "console repeater" 54 located in a power and signal distribution rack 400 receives transmissions from the control console and repeats these transmissions to one or more "trunk repeaters" 55. One such trunk repeater 55 is normally located in the distribution rack 400 with the console repeater 54.

The trunk repeater supports data communication links 26C to seven trunk cable connectors 402 which, when connected to appropriate multi-conductor trunk cables, provide power and data to repeater boxes normally hung in a lighting truss in proximity with the lamp units. The console repeater 54 also supports links 26B to eight auxiliary data connectors 404, which in turn are used to connect to additional distribution racks 408 through appropriate data input connectors 406. Data signals are thereby provided to trunk repeaters 55 located in the additional distribution racks. The additional distribution racks then provide power and data to other repeater boxes normally hung in the lighting truss in proximity with other lamp units. Each repeater box then provides power and data for up to nine lamp units.

In one embodiment of the lighting control system, one control console 24 connects to one "master" distribution rack 400, and thereafter to eight "slave" distribution racks 408 via the auxiliary data output connectors 404. Each

distribution rack connects to seven repeater boxes via the trunk cables. Each distribution rack can then provide power and data for up to 63 lamp units. One master rack and eight slave racks can then provide power and data for up to 567 lamp units. To expand the system capacity to the 1,000 lamp unit configuration supported by the system software, each slave rack 408 can connect to an additional slave rack via a data output connector 410 driven by a spare output of its trunk repeater. Eight additional slave racks so connected via link 26E provide power and data for up to 504 additional lamp units, well in excess of the 1,000 lamp units supported by the system software.

As shown in FIGURE 19, the broadcast network provides the same data signal to all lamp units practically simultaneously. Through the broadcast network 38 the console 24 sends each message to each receiver (RX) in each lamp unit simultaneously.

FIGURE 20 shows the interconnections of the reply network 40. The console 24 acquires status data from the lamp units by sending a message to the first lamp unit over the broadcast network 38 and then awaiting that lamp unit's response over the reply network 40. After the status report message has been received by the console from that lamp unit, the sequence can be continued for the other lamp units in the system. The reply network is connected in a fashion similar to the broadcast network, except that the lamp units include

transmitters (TX) for sending messages while the console includes a receiver (RX) for receiving messages.

During each reply transmission, only one of the many links 40D between lamp units and repeater boxes is utilized. As shown in FIGURE 20, a reply transmission reaches the console through only one link 40C between a repeater box and a distribution rack, only one link 40B between a trunk repeater and the console repeater, and the one link 40A between the console repeater 54 and the control console 24. Thus, if one unit of time is required to acquire status data from one lamp unit, it will take 1,000 units of time to acquire status data from all 1,000 lamp units.

It can be readily appreciated that if two or more lamp units were to respond to one request for status data, multiple transmissions would appear simultaneously on the link 40A between the console repeater and the control console. Similarly, any noise injected into the reply network would be superimposed over legitimate signals on the link 40A between the console repeater and the control console, resulting in a garbled reception by the console. Improved repeaters according to one or more aspects of the present invention provide the ability to identify and isolate erroneous lamp unit transmissions and noisy links in the reply network.

An improved repeater shown in FIGURE 21 includes a processor 450 and its associated read-only memory, random-access memory, and control circuit for receiving inputs from

the activity sensors and Manchester decoder/encoders. The processor interprets these inputs and turns on the LED's 396 and 398 by its associated control circuit to indicate the condition of the data link networks. For example, a green LED is lit to indicate a properly working data link network while a red LED is lit to indicate a malfunctioning data link network. In the absence of any activity, both LED's can be turned off. Separate pairs of red and green LED's are provided for the broadcast and for the reply data link networks. Alternatively, an alpha-numeric display device 452 may be incorporated into the repeater circuit to display simple codes or messages.

Another improved repeater shown in FIGURE 21 incorporates a multi-protocol communications controller chip 454 such as used in the console and lamp unit communications circuits. Using the communications controller chip coupled to the Manchester decoder/encoders, the processor can now detect line activity not resulting in a valid communications controller interrupt. The additional gates 456 and multiplexer 458 shown enable the processor to sample individual reply line inputs and to disable certain reply line inputs to stop noise or unauthorized transmissions from spreading into other parts of the system.

In one mode of operation, signals appearing at gates 456 are applied to a nine-input logical OR gate 460 and combined into one signal on line 462. Since only one of the nine

inputs to gates 456 will be active at any one time in a properly working system, only one signal will appear on line 462 at one time. The signal on line 462 is connected via logic gate 466 to Manchester decoder 468 and thereafter applied to communications controller 454 where it can be examined for errors by processor 450. If no errors are detected, the processor and communications controller transmit the message via Manchester encoder 470 onto the next branch of reply network 40.

If errors are detected in the signal received over reply network 40, a diagnostic mode is entered by the repeater processor 450. Using a plurality of logic control signals shown as control bus 476, the processor 450 operates multiplexer 458 to sample the various discrete signals at the input to OR gate 460. The output of the multiplexer 458 on line 464 is applied to gate 466 which is operated via control bus 476 to connect the multiplexer 458 to Manchester decoder 468.

By coordinating the operation of the multiplexer 458 with the communications controller 454 in error detection mode, the processor 450 may determine that one of the lamp units connected thereto is transmitting unintelligible signals or noise or is transmitting at inappropriate times, thereby garbling other legitimate signals. The processor then utilizes control bus 476 to disable the offending input at

gates 456, thereby restoring communication integrity for the properly functioning lamp units.

Broadcast messages are handled in a similar fashion. Signals appearing on broadcast network 38 are applied to Manchester decoder 472 and thereafter to communications controller 454 where they can be examined for errors by processor 450.

According to another feature of the present invention, a "smart repeater" shown in FIGURE 23 further includes a direct memory access (DMA) controller 480 connected between the communications controller 454 and random access memory (RAM) 482. This configuration is functionally equivalent to the processor/modem complex used in the control console and in the lamp units. One of the advantages derivable from this circuit arrangement is that each smart repeater can now communicate with the console just as any lamp unit can.

The console can send network control messages which are received by all repeater units practically simultaneously. A network control message may be addressed to a specific repeater unit or the message may be addressed to all repeater units using a common repeater address. Each repeater unit individually responds to the message depending on the address or the content of the message. For example, a message instructing the repeaters to begin status polling of the lamp units would be sent to a common repeater address. A message instructing a specific repeater to transmit a block of lamp

unit status data to the console (or to the next repeater unit along the reply network) would be sent to a specific repeater address. The repeater also sends network state messages as required, which messages include for example: data representing the kinds of errors detected, which branches of the network exhibit errors, and which branches have been disabled.

In one mode of operation, signals appearing on broadcast network 38 are detected by activity sensor 392 and decoded by Manchester decoder 472. The signals are then routed through bypass gates 484 to communications controller 454. DMA controller 480 and communications controller 454 receive the signals into RAM 482 where the decoded message can be examined or interpreted by processor 450. If no errors are detected and the message contains information for lamp units, the processor may compose a new message or re-transmit the original message to the lamp units. DMA controller 480 and communications controller 454 then cooperate to transmit the message through bypass gates 484 via Manchester encoder 474, which is coupled to broadcast network 38 by gates 478. Using control bus 476, now reconfigured to operate the gates 456 and 478, processor 450 can transmit broadcast signals to all nine outputs coupled through gates 478, or to any one or more individual output coupled thereto. Control bus 476 also operates input gates 456 so that selected individual inputs can be disabled or enabled in the manner described above.

In the reply mode, if errors are detected in the signal received from the reply network 40, the repeater unit may request the lamp unit to transmit the message again. If after several tries, the repeater cannot get an error-free message from a particular lamp unit, or if the repeater processor detects errors on two or more channels connected thereto, a diagnostic mode is entered by the repeater processor. If no errors are detected, the processor and communications controller transmit the message via Manchester encoder 474 onto the next branch of the reply network 40.

Other improvements derivable from a smart repeater include: detecting line activity not resulting in a valid communications controller interrupt; reception of framing errors, cyclical redundancy check (CRC) errors, or overrun errors detected by the communications controller; detecting errors in the header data added to each message by communications software; detecting logical errors in some of the data messages; receiving not-acknowledge (NACK) or detecting lack of an acknowledge (ACK) signal in response to transmitted messages; disabling reply line inputs to stop noise or unauthorized transmissions from spreading into other parts of the system; collecting status data from a plurality of lamp units or status data from other repeaters; and downloading operating system programs to lamp units or repeaters.

Advantages derived from the above improvements include:

1) improved feedback to the system technician, making the data link indicators easier to read and understand; 2) reporting the location of sensed errors back to the console for display at a central location; 3) improved ability of the system to operate in a degraded mode (communications errors present); and 4) improved through-put of the communications subsystem in the normal mode, especially as relates to status polling.

A smart repeater according to the present invention, as shown in FIGURE 23. enables dramatic improvements to both the utilization of communication links and the time required to collect data from all the lamp units. Once the console sends a message initiating the process, for example a message broadcast to a common repeater address, all of the truss repeaters 56 simultaneously collect data from the nine lamp units connected to each truss repeater. All of the trunk repeaters 55 then simultaneously collect blocks of data from the seven truss repeaters 56 connected to each trunk repeater 55. The console repeater 54 in master distribution rack 400 then collects blocks of data from the nine trunk repeaters 55 connected to the console repeater 54, and sends the entire block of all data collected to the console 24 in one message.

According to one embodiment, the utilization of communication links is increased because 63 truss repeaters 56 are using 63 links 26D at any one time. Only nine units of time are required to collect status data from 567 lamp units

into the truss repeaters. Thereafter, nine trunk repeaters 55 are using nine links 26C at once. Seven units of time are required to collect status data from 63 truss repeaters into the trunk repeaters. The one console repeater 54 still uses only one link 26B at a time, and requires nine units of time to collect status data from the nine trunk repeaters.

More significantly, the console receives status data from 567 lamp units in one transmission from console repeater 54 over reply link 40A, thus saving the time required to transmit 566 message headers. The same volume of data is transmitted with much less overhead. Thus, the improved lamp-to-console reply process results in drastic reductions in both the time required to collect status reports and in the probability of error. Moreover, while the lamp units are transmitting data to the truss repeaters, trunk repeaters are transmitting data to the console repeater; and while the truss repeaters are transmitting data to the trunk repeaters, the console repeater is transmitting data to the console; thereby further increasing utilization of the data links. In this way the smart repeaters interleave their own status information into the collection of lamp unit status data.

A smart repeater according to the present invention maintains operating system programs for all lamp units connected thereto and performs any necessary down-loads without tying-up the whole system. The storage and download of the operating system programs may be made depending upon

the configuration of the respective lamp units. In the case of a truss repeater performing such a down-load, only the other eight lamp units connected thereto are prevented from receiving any system cue commands during the down-load, the rest of the system being free to operate normally. Moreover, if all lamp units require operating system down-load, several smart repeaters hanging in the lighting truss can perform the operation in much less time than one control console can.

A smart repeater as shown in FIGURE 23 includes a set of gates 478 for the various broadcast link outputs and a separate set of gates 456 for the various reply link inputs. This arrangement enables the smart repeater to communicate with selected lamp units individually. If, for example, two lamp units are accidentally set to the same address, both will transmit status reports upon receipt of a request for status. This results in garbled reception at the repeater. The smart repeater then transmits to each output individually, requesting from the lamp unit connected thereto the identity or address assigned to that lamp unit, and receives the response over the corresponding input. If two lamp units are set to the same address, the smart repeater determines this to be the case and reports the information to the console for display to an operator. The smart repeaters themselves can be identified by the setting of form and function switches (to identify the processor as a repeater and not a lamp unit) and by the setting of thumbwheel switches (to identify which

repeater the processor is), both of which are included in a repeater unit identity circuit 494. Alternatively, the console repeater 54 can assign an identity to each trunk repeater 55 connected thereto, transmitting that identity via each of its nine outputs one-at-a-time. Thereafter each trunk repeater 55 can assign an identity to each truss repeater 56 connected thereto, transmitting that identity via each output one-at-a-time.

Any system utilizing processor-controlled devices must accommodate the possibility of a processor lock-up, a condition in which the processor may cease to perform its normal function due to corrupted data or the inadvertent execution of an endless loop of program instructions. The smart repeater of the present invention anticipates this possibility and provides a set of logic gates 484 associated with the communications controller, which route signals to and from the Manchester decoders and encoders and the communications controller. In a default state, the bypass gates route the output of the broadcast decoder 472 to the input of the broadcast encoder 474, while also routing the output of the reply decoder 468 to the input of the reply encoder 470. Each of the decoders and encoders themselves are connected in a default state as "repeaters", re-encoding the signal which appears on its input and providing the signal to its output. The default state of the signal repeater unit at

choices written to the display unit by the processor. This way, a technician may request error code reports, initiate diagnostic routines or other functions by communicating with the repeater unit processor through a simple, built-in data terminal arrangement.

As shown in FIGS. 17 and 18, and discussed above, the data link traffic between the control console 24 and the various lamp units travels over link 26A between the control console 24 and the console repeater 54, located in distribution rack 400. A preferred embodiment provides a duplicate link 580, shown in FIG. 17, via a connection to a backup control console 582, in case of hardware failures. This duplicate link 580 remains dormant, however, until activated, providing no additional data communications capacity beyond what the main data link 26A already provides.

In a preferred embodiment, the lighting system disclosed herein is controlled by a modular control system to facilitate the upgrade or replacement of individual system components or modules for incorporation of software and/or hardware enhancements, as necessary, without affecting the operation of the entire control system. In addition, individual modules are preferably interchangeable, so that the modular control system 500 can be reconfigured, as necessary, to accommodate the varying requirements of different shows.

As shown in FIGS. 26 and 27, the modular control system comprises a modular controller mainframe 500, interconnected

with control panel units 546-552 and other control devices in a Control Resources Network by a set of input modules 590, for controlling lamp units in a Device Control Domain interconnected with the modular controller mainframe 500 by a set of output modules 592.

Controllers for modern lighting systems often must be capable of simultaneously supporting diverse lamp units having different communication protocols, functions and data requirements. For example, lighting designers often desire to incorporate conventional (intensity-only) lamp units, in addition to automated variable-parameter lamp units, as well as utilizing lamp units provided by different manufacturers.

While conventional lamp units only require an intensity data value, automated variable-parameter lamp units, such as those associated with the Vari*Lite® Series 200™ lighting system, will require a number of variable parameters including, e.g., color, intensity, gobo, pan and tilt.

Additionally, more complex automated variable-parameter lamp units capable of projecting images, e.g., lamp units having a liquid crystal display, such as those disclosed in the above incorporated application serial no. 07/693,366, require image data files in addition to the parameter data associated with a typical automated lamp unit.

Accordingly, each module, in the sets of input and output modules 590, 592, discussed further below, is configured as an independent data network, capable of conforming to one or more

different communication protocols for communicating with the specific devices attached thereto; thereby facilitating the connection of the modular controller mainframe 500 to a number of otherwise non-compatible devices. In this manner, the modular controller mainframe 500 serves as an interface system for communications between a plurality of control devices having diverse communications protocols and a plurality of lamp units and other output devices having diverse communications protocols, functions and data formats. Accordingly, improved control of various types of lamp units having diverse functions and protocols may be achieved.

Each module in the set of input modules 590, shown in FIGS. 26 and 27, and discussed further below, serves as an interface between the modular controller mainframe 500 and one or more control devices, e.g., 556-558, 560, 568, 570, attached thereto. Each module in the set of input modules 590 is configured to receive control commands from the control devices connected thereto and to translate, if necessary, the received commands into a command format that may be interpreted by the modular controller mainframe 500.

Each module in the set of output modules 592, discussed further below, serves as an interface between the modular controller mainframe 500 and one or more types of lamp units and other output devices having diverse communications protocols, functions and data formats. The individual modules in the set of output modules 592 receive generic console

set of output modules 592, wherein, if necessary, the output module performs the command translation before transmitting the commands to the lamp units.

For example, if a console operator adjusts a knob associated with intensity control for a number of lamp units selected for manual control, the modular controller mainframe 500 will transmit a command to each module in the set of output modules 592 consisting of an encoded representation of the "delta value" corresponding to the knob adjustment. Those modules in the set of output modules 592 that support lamp units capable of directly interpreting this generic console command, will transmit the console command to each connected lamp unit, without translation. However, those modules in the set of output modules 592 that support lamp units requiring absolute parameter commands must translate the generic console commands to the absolute parameter values necessary to set the selected lamp units to the desired intensity.

Those modules in the set of output modules 592 that are configured to perform command translations preferably include a processor and memory for storing cue data in order to duplicate the functions of a lighting controller, such as those described in U.S. Patent No. 4,392,187, to Bornhorst, wherein the controller receives commands representing manipulation of console controls, and calculates the absolute parameter values necessary for transmission to each lamp unit.

In addition, each output module 592 may be configured to conform transmitted signals according to the appropriate communications protocol for the connected lamp units, i.e., each output module ensures that transmissions to the lamp units have the appropriate signal levels, timing, parameter order and other format factors that are expected by the lamp unit.

MODULAR CONTROLLER MAINFRAME/MAIN PROCESSOR KERNEL

The modular controller mainframe 500, shown in FIG. 26, includes a main processor kernel 502 and sets of input and output modules 590, 592, all interconnected by high-speed parallel data busses, including input bus 512, output bus 572, and mass storage bus 504.

The main processor kernel 502 includes a microprocessor (CPU), such as a Motorola MC68040, random-access memory (RAM), read-only memory (ROM), and associated support circuits. The main processor kernel 502 could be constructed as a mother board having its CPU, memory (RAM and ROM) and support circuitry built thereon, with the mother board additionally providing built-in connectors for mating with plug-in connectors of the various buses 504, 512, 572.

The main processor kernel 502 communicates in one of several modes with the various modules in the sets of input and output modules 590, 592 for the transfer of cue data, status reports and other information. In a manner similar to the communications manager program, described above for

communications between the console and the individual lamp units, there can be provided two types of message addresses for communications between the main processor kernel 502 and the various modules in the sets of input and output modules 590, 592; namely, individual module addresses and a module broadcast address.

Accordingly, each module may be individually accessed by the main processor kernel 502 by transmitting the unique module address associated with that particular module. In this manner, although all modules can receive the transmission with the individual module address, only the module associated with the transmitted address will respond.

In the module broadcast mode, the main processor kernel 502 can transmit messages to all modules in the sets of input and output modules 590, 592 at a common address, wherein each module in the sets of input and output modules 590, 592 responds depending upon the respective configuration of the module, and the devices connected thereto. In one embodiment, additional module broadcast addresses can be provided to allow the main processor kernel 502 to limit a broadcast to either the set of input modules 590 or the set of output modules 592.

In a lamp unit broadcast mode, control input signals received by the modular controller mainframe 500 from any of the control input devices are transmitted via the set of output modules 592 to each of the lamp units connected in the Device Control Domain at a common address.

|||

MASS STORAGE DEVICES

The main processor kernel 502 communicates, via the mass storage bus 504, with processor-controlled mass data storage and retrieval devices, e.g., a large capacity hard disk drive 506 or a digital data tape cartridge drive 508. In addition, a floppy disk drive 510 may be connected directly to the main processor kernel 502.

Preferably, the mass storage bus 504 conforms to the Small Computer System Interface (SCSI) protocol, or a similar bus standard, so that additional mass storage devices (not shown) can be easily connected to the mass storage bus 504.

Cue data uploads, received from the lamp units by the output modules 592 and then transmitted to the main processor kernel 502, can be stored on the hard disk drive 506 in addition to being "backed-up" on the tape cartridge drive 508 and/or floppy disk drive 510. Status reports, described further below, can be logged on the hard disk drive 506 for analysis by a data base and report generator program which may operate on the main processor kernel 502 or on a personal computer 560 (FIG. 27).

OPERATING SYSTEMS

Operating system software for the various devices of the modular control system, e.g., main processor kernel 502, the sets of input and output modules 590, 592, control panel units 546-552, lamp units, and smart repeaters 52-58, can be stored on hard disk drive 506. In this manner, updated versions of

the operating system software can be loaded onto the hard drive 506 via a tape cartridge inserted in the tape drive 508, or via floppy disks inserted in the floppy disk drive 510. Operating system software can thereafter be downloaded from the mass storage devices 506-510 to the appropriate devices, as required.

INPUT MODULES

The main processor kernel 502, as noted above, is connected to the set of input modules 590, each controlled by a microcontroller integrated circuit, such as the Motorola MC68302 Integrated Multiprotocol Processor. The set of input modules 590 are connected to the main processor kernel 502 by means of parallel input bus 512, e.g., a 16-bit or 32-bit data path, also having associated address and control lines, as required.

The configuration of the input modules 590, shown in FIG. 26 and discussed below, is merely illustrative. Other input modules may be connected to the input bus 512, as required, to accommodate new control devices and serial data link formats, which may be developed in the future.

Input module 514 connects control panel units 546-552 to the modular controller mainframe 500, as shown in FIG. 27, via a well known serial token bus 523, or similar serial data link. Preferably, input module 514 is embodied as a token bus controller, for communicating with microprocessor-controlled control panel units 546-552. A microprocessor station.

connected to the serial token bus may transmit on the bus when it has possession of the control packet, or "token", with control being surrendered upon transmission to the processor receiving the control packet. Alternatively, a token ring network may be implemented, with the various devices on the network connected in a daisy-chain fashion to form a closed ring.

Input module 514 is preferably configured to "listen to" all messages on the serial token bus. In this manner, input module 514 can transmit at any time input control signals to the main processor kernel 502, where appropriate, regardless of which processor station connected to the bus 512 possesses the control packet.

The control panel units 546-552 are each controlled by one or more microprocessors and may be configured to incorporate particular features and functions of a control console, described above, such as the Artisan® console marketed by Vari-Lite, Inc., of Dallas, Texas, in addition to supplementary features.

For example, a manual control panel unit 548 can provide, e.g., means for selecting lamp units to be placed under manual control, means for indicating selected or active lamp units, and means to manipulate the various parameters of selected lamp units. In a typical lighting arrangement, manual controls are utilized for arranging the various lamp

units in a desired configuration prior to storing the resulting lighting effect as a "cue".

In addition, a submasters control panel unit 550 can provide, e.g., means for storing, recalling and initiating cues, as well as means for executing manual cross-fades between two cues. Once a cue has been assigned to a submaster, it may typically be activated by selecting the associated submaster and manipulating the fader on that submaster.

A chase/matrix control panel unit 552 may provide means for controlling chase sequences, in addition to matrix control means for controlling smaller groups of lamp units. A chase is a programmed sequence of cues that may be recalled into the chase/matrix control panel unit 552 and executed automatically in a known manner. A known matrix panel provides faders for manipulating the intensity of lights in a recalled cue, provided that the lights have previously been "patched" to the matrix panel.

An additional control panel unit 546, featuring a subset of the manual controls, cue storage and recall, and chase and matrix controls described above, may be used as an alternate control console in a remote location or as a backup control console. This remote or backup console may be configured to incorporate particular features and functions of a remote or backup control console, such as the mini-Artisan® console marketed by Vari-Lite, Inc., of Dallas, Texas, in addition to

supplementary features. The control panel unit 546 is preferably constructed on a single control panel.

Input module 514 is preferably configured to receive and sample data at a sufficiently high data rate such that each of the control panel units 546-552 connected to the serial token bus 523 can be on-line simultaneously. This allows a remote operator to perform control operations on the remote control panel unit 546 while the main operator performs operations on one or more of the control panel units 548-552.

Each control panel unit 546-552 preferably includes one or more display modules, which may be liquid crystal displays (LCD), electro-luminescent (EL) graphic display panels, vacuum-fluorescent (VF) alphanumeric display modules, light-emitting diode (LED) character display elements, or other suitable display devices.

The control panel units 546-552 may also include "soft switches", for example, push button switches having character display means in the key cap of the push button, such as the Pixie Graphic LCD Switch made by Industrial Electronic Engineers, Inc., of Van Nuys, California. Alternatively, push button switches can be mounted adjacent to other display units which indicate the functions of the buttons.

In this manner, as the function of the push button changes in different operating modes or under control of different operating system software versions, the label

displayed in the key cap or in the adjacent display can be re-written to indicate changed or alternate functions.

Some of the push button switches may be fixed-function, for example a numeric keypad, while other buttons are reprogrammable in different modes of operation, for example a bank of buttons might be channel select buttons in one mode, and timing function controls in another mode.

The control panel units 546-552 may also contain one or more types of continuous manual control devices, for example rotary knob-type controllers, such as rotary optical encoders, or linear fader-type controllers such as linear slide-potentiometers. A manual control panel 548 may contain only rotary continuous controllers while a submasters control panel 550 may contain only linear continuous controllers. A special function control panel such as a chase/matrix controller 552 or a remote/backup console 546 may contain both rotary and linear controllers.

The primary control panel units 548-552 may be mounted together in a single console 554 to serve as a main control console or, alternatively, they may be mounted separately in spaced-apart locations. Further, each of the control panel units 546-552 can be customized for a particular application by installing the desired mix of knobs, faders and display units, and programming the soft switches, as desired.

The modular design of the control panel units 546-552 allows the controller to be designed in accordance with

prevailing desires of lighting system operators, which can thereafter be easily supplanted by a different configuration if those desires change, without requiring a complete redesign of the hardware and operating system programs of the entire lighting control console. Since each control panel unit 546-552 runs its own operating system programs, the necessary program modules can be re-written to support a new control panel unit while the rest of the modular control console system remains unchanged.

Additional input modules can be provided as well, as shown in FIG. 26, including, e.g., a Musical Instrument Digital Interface (MIDI) module 516, an Ethernet port module 518, an RS232 serial data port module 520 or a video input module 522.

The MIDI input module 516 is configured to receive and interpret signals that conform to the MIDI convention. In one embodiment, the MIDI input module 516 may be connected to a MIDI recorder/sequencer 556, such as the Alesis MMT-8 Multi Track MIDI Recorder, by means of a serial data link 524, for recording and playback of control console commands generated by the modular controller mainframe 500, as shown in FIG. 27 and discussed further below.

In another embodiment, discussed further below, the MIDI input module 516 may be connected to an electronic musical instrument 557 or other device capable of generating MIDI "notes", with the MIDI notes generated thereby received and

interpreted by the modular controller mainframe 500 to effectively control lighting parameters. The electronic instrument 557 may be connected to the input module through the MIDI recorder 556, as shown, or alternatively, the instrument 557 may be coupled directly to module 516.

Since the MIDI convention defines 128 MIDI notes, up to 128 buttons on the various control panel units 546-552 may be "mapped" to unique MIDI notes. Accordingly, upon depression of a mapped button on a control panel unit 546-552, a console command will be detected by the main processor kernel 502, whereupon the main processor kernel 502 can translate the console command into the "mapped" MIDI command. The MIDI NOTE ON command may be then transmitted to the MIDI input module 516 for recording by the MIDI recorder 556. Similarly, when the button is released, a MIDI NOTE OFF command may be transmitted to the recorder 556.

By configuring the recorder 556 to record events received at its MIDI IN port, the series of commands generated by button depressions on the control panel units 546-552 may be stored in the non-volatile memory of the recorder 556 for later retrieval.

In this manner, "Board Control Cues" may be established that place the control panel units 546-552 in a particular condition for execution of a series of cues for a particular song or dance. For example, to prepare the submasters control panel 550 for a particular song, the associated cue numbers

for that song or dance must be loaded into specific submasters and the appropriate submasters must then be selected. Similarly, to prepare the chase/matrix control panel 552 for a particular song, certain chases and matrix patches must be loaded.

The console operator typically performs these tasks during a performance, prior to each song, by executing a series of button depressions on the control panels. Unfortunately, the order of songs to be performed is often not known in advance, and the operator will only have a minimal amount of time, following notification of the next song to be performed, to arrange the control panels 546-552 in the necessary configuration.

Accordingly, the MIDI recorder 556 can record for each song (in advance) the series of button depressions necessary to place the control panel units 546-552 in the particular condition for the series of cues associated with each song or dance. Thereafter, during a performance, when the operator is notified of the next song to be performed, the operator can initiate the playback of the previously recorded button depressions by the MIDI recorder 556. The MIDI recorder 556 effectively performs the quick burst of key depressions required to setup the control panel units 546-552 in a particular condition.

The MIDI signals received at the MIDI input module 516 from the MIDI recorder 556 during playback are preferably

translated to the corresponding console control commands by the processor of the MIDI input module 516, prior to transmitting the commands to the main processor kernel 502 for subsequent transmission to the control panel units 546-552. Performing the translation in the MIDI input module 516 relieves the processor of the main processor kernel 502 of performing this task, and allows the main processor kernel 502 to remain free to perform other tasks.

Although, as stated above, the MIDI recorder 556 may be embodied as a multi-track MIDI recorder, capable of playing one or more tracks simultaneously, only a single track should be utilized for recording and playback so that NOTE ON and NOTE OFF commands are not directed to the same button simultaneously. In addition, it should be noted that depression of unmapped buttons on the control panel units 546-552 will not be recorded by the MIDI recorder 556.

In an alternate embodiment, the MIDI input module 516 may be configured to receive light parameter control commands generated by an electronic musical instrument 557 or other device capable of generating MIDI notes. By mapping MIDI notes to parameter control commands, as described above, lighting system parameters may be directly altered from a remote source by depressing the "keys" corresponding to the desired "notes" on the MIDI instrument.

The MIDI NOTE ON and NOTE OFF commands generated by the MIDI device 557 will be communicated to the lighting system

via the MIDI input module 516, where the MIDI commands will preferably be translated to the corresponding console control commands by the processor of the MIDI input module 516, before transmission to the main processor kernel 502.

Commands received at the modular controller mainframe 500 for controlling the state of a console button can be interpreted in either of two ways, i.e., to toggle the state of the button to its alternate state or to place the button in a desired state, regardless of its prior state. Accordingly, two modes are preferably provided. In the first mode, parameter control commands received by the modular controller mainframe 500, i.e., a MIDI NOTE ON or NOTE OFF command, will be interpreted as a command to toggle the associated console button to its alternate state. Accordingly, if a MIDI NOTE ON command is received in this first mode, corresponding to a button already selected, the NOTE ON command will toggle the button to its alternate, or deselected state.

In the second mode, however, commands received at the modular controller mainframe 500 are interpreted to place the associated buttons into a known state regardless of the prior state of the button.

For example, a message received by a module in the set of input modules 590 may contain a command to select certain submasters and/or deselect others, regardless of whether or not the submasters are currently selected when the message is received by the input module.

Upon receipt of this command by the modular controller mainframe 500, the main processor kernel 502 will recognize the commands to select and/or deselect certain submasters. The main processor kernel 502 will recognize that this message should be directed to control panel units 546-552 and will subsequently send a message containing the commands to input module 514 for transmission to the control panel units 546-552.

The control panel units 546-552 will each receive the message and individually respond, depending upon whether or not the specified submasters reside on the individual control panel unit. The control panel units having the specified submasters will respond to the command by lighting the appropriate select button [SEL], if not already lit, as an indication that the associated submaster is selected. The control panel units not having the specified submasters will ignore the submaster select command.

In addition, messages containing the submaster select/deselect commands should also be sent by the main processor kernel 502 to each of the various output modules in the set 592 for transmission, where appropriate, to each of the lamp units. As discussed above, for those lamp units not capable of interpreting "generic" console commands, the associated output modules 592 must translate the commands into the absolute parameter data, suitable for interpretation by the lamp unit.

Input module 518, shown in FIG. 26, implements an Ethernet port for high-speed data communications. In one implementation, the Ethernet port may be connected to a graphics workstation 558, such as a Sun Microsystems SPARC 2 computer workstation, via serial data link 526, as shown in FIG. 27. A graphics workstation 558 may be utilized to develop, modify and control images which are to be projected by automated lamp units having image generating capabilities, such as the lamp units having liquid crystal projection gates disclosed in the above incorporated application serial no. 07/693,366.

In addition, graphics workstation 558 may be utilized to operate software suitable for coordinating the off-line programming of lighting parameters by utilizing a three-dimensional model of the performance venue and the functions of a lighting system, such as the programming and modelling tool described in the above incorporated application serial no. 07/641,031.

Input module 520 preferably implements an RS232C-compatible serial data port. In one embodiment, shown in FIG. 27, the RS232 port can be connected to a personal computer 560 via serial data link 528, allowing cue data, status reports and other information to be transferred between the various lamp units, control panel units 546-552 and personal computer 560. In this manner, the personal computer 560 can be

utilized for development, display and manipulation of cue data and status reports.

Preferably, a suitable modem circuit is included in input module 520, since the RS232 data format is commonly transmitted over telephone lines. As shown in FIG. 27, the modular controller mainframe 500 can thus communicate with remote devices and computer systems via serial data link 530 and telephone line interface 562, thereby allowing cue data, status reports and other information to be transmitted between the performance venue and a remote maintenance facility. In this manner, faults in the lighting system may be diagnosed by an operator at the remote facility.

Input module 522 is preferably configured to receive analog video inputs via serial data links 532 from, e.g., a video tape recorder 570 and/or video camera 568, as shown in FIG. 27. In this manner, video signals generated by the recorder 570 and/or camera 568 may be multiplexed and transmitted to those lamp units having image generating capabilities via separate data link 584 (discussed below).

OUTPUT MODULES

As noted above, a set of output modules 592 are preferably provided for interfacing the main processor kernel 502 with the lamp units. The set of output modules 592 are connected to the main processor kernel 502 by means of parallel output bus 572, e.g., having a 32-bit or 64-bit data path, in addition to the necessary address and control lines.

Each output module in the set 592 is preferably controlled by one or more microprocessors, such as the Motorola MC68302 Integrated Multiprotocol Processor and the Motorola MC68332 microcontroller.

As noted above, each module in the set of output modules 592, is preferably configured as an independent data network, allowing each module to serve as an interface between the modular controller mainframe 500 and one or more types of lamp units and other output devices having diverse communications protocols, functions and data formats. Each output module in the set 592 translates, if necessary, the generic console commands received from the modular controller mainframe 500 into the specific commands or parameters necessary for communicating with the specific types of lamp units or output devices connected thereto.

The configuration of the modules in the set of output modules 592 and the lamp units or other output devices connected thereto, e.g., the number and type of lamp unit connected to each output module, as shown in FIGS. 26 and 27 and discussed below, is merely illustrative, with other output module configurations being easily developed, as necessary, to accommodate varying preferences in the variety, number and arrangement of lamp units and other output devices comprising each lighting system. Furthermore, additional individual output modules can be constructed or upgraded, as necessary, to accommodate any new communication protocols, functions or

data formats that may be developed for lamp units and output devices.

As noted above, each output module in the set 592 must be able to conform signals, prior to transmission to the connected lamp units, to the appropriate communications protocol, i.e., each output module must ensure that transmissions to the lamp units have the appropriate signal levels, timing, parameter order and other format factors that are expected by the lamp unit.

In the illustrated embodiment, the lighting system consists of automated variable-parameter lamp units capable of directly receiving generic console commands, e.g., those associated with the Vari*Lite® Series 200™ lighting system (VL200S), such as the VL2®, VL2B®, VL3™ or VL4™ luminaires; automated variable-parameter lamp units requiring absolute parameter commands, i.e., those requiring translation of the generic console commands; automated variable-parameter lamp units capable of projecting stored images; automated variable-parameter lamp units capable of projecting video images; standard conventional, fixed-focus lamp units and other output devices, e.g., air cannons, special effects projectors, smoke machines and chain hoist motors for controlling the elevation of the truss assembly.

Preferably, as noted above, each module in the set of output modules 592 that support lamp units capable of reacting to received commands is capable of transmitting commands on a

broadcast basis to a common lamp unit address and to transmit lamp-specific commands to individual lamp unit addresses.

In addition, the set of output modules 592 is preferably configured to download operating system software from the mass storage devices 506-510 to individual lamp units and smart repeaters connected thereto, as required. Each module in the set of output module 592 may be programmed to accomplish other tasks, as well, according to the operating system software written for that output module.

In the illustrated embodiment, output module 534 is configured to support automated variable-parameter lamp units capable of directly receiving generic console commands, such as those lamp units associated with the Vari*Lite® Series 200™ system. When output module 534 is configured to support the communications protocol of the Vari*Lite® Series 200™ system or a similar protocol, it preferably implements bidirectional serial data link 26A2 using Manchester encoding, as described above, which features separate broadcast and reply data links in a dual network.

Output module 535 is configured to support automated variable-parameter lamp units requiring absolute parameter values. Accordingly, output module 535 must be capable of translating the generic console commands received from the main processor kernel 502 into the absolute parameter values required for the specific types of lamp units connected thereto. As noted above, output module 535 preferably

includes a processor and memory for storing cue data in order to perform the necessary command translations.

In the illustrated embodiment, output module 536 is configured to support automated variable-parameter lamp units capable of projecting images, e.g., those having liquid crystal projection capabilities. Accordingly, output module 536, connected to the associated lamp units by means of data link 26A4, is preferably configured to interleave stored digital data files with the system commands and parameter data typically downloaded by the controller for storage in the local memory of the lamp units prior to a performance. The image data files may be received from graphics workstation 558 or from a mass storage device 506-510.

The image signals associated with output module 536 are typically digital image signals; accordingly, these signals may be carried between the output module 536 and repeater 54A4 for transmission to the connected lamp units by means of a dedicated twisted pair 574, which is typically utilized in lighting systems.

Output module 538 may be configured to support automated lamp units capable of projecting video images, e.g., those lamp units having liquid crystal projection capability. Accordingly, output module 538, implementing data link 26A5, may transmit analog video signals to the connected lamp units. The video signals may be received through input module 522

connecting a source of analog video, e.g., camera 568 or video tape recorder 570, as discussed above.

Since the video signals associated with output module 538 are typically analog signals they may be carried between the output module 538 and the connected lamp units by means of a dedicated coaxial cable 576. Output module 538 may be reconfigured at such time as necessary to accommodate digital video signals.

The lamp units having liquid crystal projection capabilities can be configured to generate animated pictures by sequencing stored digital image data files. Accordingly, output modules 536 and 538 are preferably configured to transmit image files to these lamp units in a real-time sequence. Alternatively, the image data files can be transmitted to the lamp units for storage in advance of a performance and recalled in a sequential manner similar to the retrieval of cue data. For those video lamp units requiring analog video signals, the digital image data files for sequential projection are preferably converted to analog signals by output module 538 prior to transmission to the lamp units.

Output modules 536 and 538, supporting image generating lamp units, preferably include the appropriate hardware for image handling, i.e., additional memory capacity for storage of image files, and conversion circuitry, if necessary, for converting image files from analog to digital signals, or

vice-versa, as required by the type of lamp units connected thereto. In addition, these output modules may require enhanced transmission capacity for transmission of the image data files, i.e., a high-speed optical-fiber data link or, alternatively, compression circuitry for transmission of real-time animated sequences over a relatively slow data link.

In the embodiment shown in FIG. 26, a separate data link 584 is provided between video input module 522 and video output module 538 for transmitting video signals therebetween. In the case of analog video signals, the additional link 584 allows the video signal to be transmitted from the input module 522 to the output module 538 as an analog signal, without requiring analog-to-digital conversion in module 522 and then reconversion from digital-to-analog in module 538. Additional link 584 is preferably utilized even where digital video signals are utilized, to prevent overburdening the transmission capacity of main processor kernel 502 and internal buses 512 and 572.

As shown in FIG. 27, video input module 522 is preferably configured to accept several video inputs simultaneously. Additionally, each of the video signals generated by these video inputs are preferably multiplexed onto line 584 and provided to output module 538 for subsequent transmission to each of the connected video lamp units. In this manner, each of the video lamp units is provided with a number of video signals for projection. Each video output module 538 is

preferably configured to translate each of these video signals into a format that is compatible with each of the video lamp units connected thereto.

The digital image files are preferably broadcast by the main processor kernel 502 to each of the output modules. However, only those modules in the set of output modules 592 supporting image generating lamp units should respond.

Control signals specifying certain changes to the image data at the lamp unit projection gates can be transmitted from the modular controller mainframe 500 to the lamp units to cause the lamp unit processor to execute such functions as a video "dissolve" or "wipe" from one image to another. Such "dissolve" or "wipe" commands may originate from a graphics workstation 558 or from a control panel 546-552, in the manner described with respect to bidirectional bus 80 in the Control Resources Network, as shown in FIG. 2.

In the illustrated embodiment, output module 540 implements a DMX-512 serial data link 542 for controlling ac power dimmers 544. Although the DMX-512 protocol supports up to 512 channels for conventional, fixed focus lamp units, less than 512 channels can be allocated to conventional lamp units.

In the illustrated embodiment, output module 533 is configured to support a plurality of output devices, other than lamp units, commonly utilized in performances. As described above, other stage action effects often need to be controlled by a lighting console. For example, output module

533 can be configured in a manner similar to the control signal converter 64, described above with respect to FIG. 2, for producing control signals for directing the operation of chain hoist motor 66, air cannon 68, special effects projector 70 and a smoke machine.

STATUS REPORTS

Each module in the set of output modules 592 that supports lamp units capable of transmitting status reports is preferably configured to receive such status reports from the lamp units for transmission to the main processor kernel 502. Thereafter, the status reports can be transmitted by the main processor kernel 502 over the input bus 512 to each module in the set of input modules 590 for transmission to the control devices.

For example, status reports received by the main processor kernel 502 can be relayed to control panel units 546-552 via input module 514 and/or to personal computer 560 via input module 520. Preferably, the operating system software of these control devices, e.g., 546-552, 560 allows the control devices to receive and display such reports.

In addition, an artificial intelligence program or "expert system" may be installed as part of the operating software of these control devices 546-552, 560 for monitoring and analysis of the status reports. In this manner, the control devices can scan the status reports, and identify

faults in the system by comparing the status reports against a data base of known symptoms and possible faults.

It thus becomes possible, for example, for the control devices 546-552, 560 to conclude and report that the ambient temperature in the vicinity of the lamp units is probably higher than normal if a preponderance of lamp units report over-temperature conditions in their respective lamp head assemblies.

Upon receipt of a status report from the main processor kernel 502, the input module 514, preferably embodied as a token bus controller, transmits the status report, along with the control packet, to one of control panel units 546-552. The first control panel unit to receive the message, i.e., unit 548, displays the report to the operator. The processor of control unit 548 will compile a message consisting of the control packet, the status report, and any control input signals generated by operator control actions, for transmission to a second control panel unit, i.e., unit 550.

The second control panel unit 550 receives the status report and displays it. The processor of second control unit 550 compiles a message consisting of the control packet, the status report, control input signals added by the first control panel unit 548, and control input signals generated by operator control actions at the second control panel unit 550, for transmission to the next device on the serial token bus 523, i.e., control panel unit 552.

Eventually, the messages are returned to the token bus controller 514. Token bus controller 514 will thereafter discard the status report and compile the control input signals for transmission to the main processor kernel 502.

Experience has shown that certain situations require different controls for operating a computer controlled lighting system with distributed processing. In a rehearsal mode, for example, extensive controls for selecting lamp units and manipulating their multiple parameters are required for programming the system to achieve the desired lighting effects. In a performance mode, on the other hand, extensive controls for recalling cue data and controlling groups of lamp units are required for operating the system to reproduce and/or combine the desired lighting effects.

In a synchronized performance mode, no controls are required as the modular controller mainframe 500 receives timing signals via one of the modules in the set of input modules 590 and recalls cues according to a time code program so that the desired lighting effects are reproduced in coordination with other events in a show.

In yet another performance mode, it may be appropriate to provide simple display means and manual override or emergency controls on a control panel unit 546-552 connected to the modular controller mainframe 500 so the desired lighting effects can be reproduced even if the timing signals are lost.

The present invention therefore provides a control system for computer controlled lighting systems with distributed processing in which the control console can be reconfigured and/or replaced with a different console depending upon the specific application for which the lighting system is used. It is no longer necessary to provide one console having all possible controls available all of the time, rather the control system can be reconfigured to accommodate the differing requirements of rehearsal programming and performance playback, including various requirements to provide electrical control interfaces between the lighting control system of the present invention and other electrical control systems.

Although several embodiments of the invention have been illustrated in the accompanying Drawings and described in the foregoing Detailed Description, it will be understood that the invention is not limited to the embodiments disclosed, but is capable of numerous rearrangements, modifications and substitutions without departing from the scope of the invention.

In another embodiment, the lighting system disclosed herein is controlled by a distributed control system to facilitate more efficient utilization of diverse control system elements. Whereas the modular control resources network described above provides for multiple on-line control panels operating simultaneously to control a plurality of automated and conventional lamp units, the modular controller mainframe remains a single, complex node of data link communications between the controllers and the lamp units. The complexity of the modular controller mainframe makes it susceptible to failure and its role as a central node of communications makes its failure a serious potential liability. To relieve the seriousness of a potential failure in a central component of a control system such as the control resources network described herein, a distributed control system characterized by distributed control system elements interconnected by a simplified high-speed data link network is utilized.

Referring now to figure 28, a distributed control system consists of one or more control consoles 24A, 24B, 24C of the type hereinbefore described, one or more general purpose computers 560A, 560B, 560C, 560D, and one or more

load interface modules 602, 610A, 610B, 610C typically but not necessarily associated with a power and signal distribution rack 400 (figure 17); all of the above described control system elements communicating over a high-speed data distribution network 600, such as Ethernet.

The various control system elements may exist as peers on the network, which is to say that no one element is in control of data communications on the network but all elements have equal access to the network; some well-understood scheme being utilized to resolve or otherwise obviate conflicts arising from simultaneous attempts to transmit data by two or more control system elements. Alternatively, a token bus such as described in connection with figure 27 may be utilized.

Other configurations may be utilized in which a control console and a personal computer are connected together locally, and one or the other is connected to the high-speed control resources network 600.

One system element has unique status as a system data repository, which may be a general purpose computer 560 that stores a three-dimensional model of the lighting system such as described in the above incorporated application

serial number 07/641,031 (now U.S. Patent No. 5,307,295) entitled "Creating and Controlling Lighting Designs."

Figure 29 illustrates typical architecture for a control console. Control consoles 24 are optimized for direct control of lamp units in the lighting system, including but not limited to manual control of variable lighting parameters, storing parameter data to be recalled later, and directing the lamp units to recall and conform to the stored parameters. Consoles exist primarily to send control commands to lamp units, and typically include limited display capabilities. A control panel 664 includes a variety of switches, knobs, faders, and displays used to direct operation of the system by a human operator. A microprocessor 650 connected to address, data, and control signal bus 654 executes programs stored in a Read-Only Memory (ROM) 652 and stores data related to the operation of the system in Random Access Memory (RAM) 652, the RAM and ROM also being connected to bus 654. Microprocessor 650 reads the control panel 664 via a control interface circuit 665, also connected to bus 654. A touch-sensitive display 662 is connected to a display interface circuit 660 connected to bus 654. The microprocessor reads touch inputs

from touch-sensitive display 662 via interface 660, and writes display data to the display via the same interface. A high-speed data link transceiver 658 connects to bus 654 and serves as an interface between the console and the high-speed control resources network 600.

Figure 30 illustrates typical architecture for a general purpose computer. Microprocessor 620 executes programs stored in ROM 622, loads programs stored on hard disk drive 625 into RAM 623 and executes the programs from RAM. Address, data, and control signal bus 624 interconnects the microprocessor, RAM, ROM, hard disk drive and other components described below. A removable disk drive 626, such as a magneto-optical hard disk cartridge, stores backup copies of the show file in a transportable form. Display interface 630 connects to bus 624 and supports operation of a graphic display device 632. High-speed data link transceiver 628 connects to bus 624 and serves as an interface between the computer and the high-speed control resources network 600. Input devices may include a standard keyboard 642 connected through interface 636; a pointing device 644 such as a mouse, light pen, trackball or similar device connected through interface 638

and; a voice recognition module 640 with microphone, connected through interface 634.

General purpose computers 560 are utilized to store and display data, including tabular views of parameter data and graphical views of a performance environment or other lighting environment. One general purpose computer in a system is a system data repository storing a "show file" containing a three-dimensional model of the environment, including lamp units and objects to be illuminated, the show file also containing parameter data for the lamp units. The show file serves as backup data for parameter data sets (cue data) stored in the lamp units. The show file can also serve other purposes as will be described herein. As described below, copies of the show file may exist on other general purpose computers in the system and means may be employed to update the copies as the show file on the system data repository is changed. Any general purpose computer in the system can display lamp unit status information and view, create, or edit any show file on the repository. General purpose computers can also run programs that create a virtual control console in a "window" on the computer's display, and allow operators to control lamp

units from the computer utilizing the window in conjunction with a pointing device 644 and/or keyboard 642.

Figures 31 and 32 and illustrate typical architecture for load interface modules. Load interface modules 602, 610 exist primarily to funnel commands to and report the status of all lamp units connected thereto. Load interface modules periodically transmit status data over the high-speed control resources network for display by control consoles and general purpose computers. Load interface modules are also network gateways to multiple device control networks, transmitting and receiving data in one format over the high-speed control resources network while transmitting and receiving data in another format over individual device control networks.

Each load interface module 602, 610 is configured as an independent data network, allowing each load interface module to serve as an interface between the high-speed distributed control resources network 600 and one or more types of lamp units having diverse communications protocols, functions and data formats. Each load interface module translates, if necessary, generic console commands received from control consoles or general purpose computers into

specific commands or parameters necessary for communicating with specific types of lamp units or output devices connected thereto.

Load interface module 602 (figure 31) includes a microprocessor 670, ROM 672 containing executable programs for operating the microprocessor, RAM 673 for storing data related to operation of the interface, a data link transceiver 678 for connection to the high-speed control resources network 600, front panel controls and indicators 676 connected to an interface circuit 675, and lamp unit interface 680 that supports an independent device control network. Address, data, and control signal bus 674 interconnects the major functional blocks of the load interface module 602.

Load interface module 610 (figure 32) corresponds to the Smart Repeater unit shown in figure 22, but is modified to communicate with control consoles over the high-speed data link 600. The Manchester Encoder 470 and the Manchester Decoder 472 shown in figure 23 are replaced in the load interface module 610 by a high-speed data link transceiver 612 for connection to the high-speed data link 600. Alternatively, both uplink interfaces can be included on one

circuit card assembly, the encoder 470 and decoder 472 as well as transceiver 612, so that one standard circuit card assembly can serve as Smart Repeater 56 (figures 24 and 25) or as load interface module 610 (figure 28).

The high-speed control resources data network 600 is preferably an industry-standard, commercially-available, local-area-network format configured as a hub unit 604, 605, 606 having ports for connection to devices on the network. Transceiver circuits 628, 658, 678, 612 configured as plug-in circuit card assemblies are installed in each console, general purpose computer, and load interface module. Alternatively, transceiver circuits can be built into other circuit card assemblies designed to implement the functions of the control console or load interface module, but a plug-in card configuration allows the transceiver circuits to be easily removed and replaced by improved transceiver circuits if and when an improved network format becomes available.

One hub unit 604 is preferably co-located among control consoles and general purpose computers positioned, for example, in a seating area adjacent to a stage or other performing area, or in a lighting control booth. Control consoles 24A, 24B and general purpose computers 560A, 560B

are connected to the hub unit 604 which is then connected to a second hub unit 605 co-located among power and signal distribution racks containing load interface according to the present invention. Load interface modules modules 610A, 610B, 610C connect to the second hub unit 605 and provide control signals generated by consoles and general purpose computers to lamp units connected thereto. A third hub unit 606 connected to the second hub unit 605 may be provided to support larger numbers of lamp units. Connections between hub units may be high-bandwidth links such as fiber-optic cable communications format while connections between hub units and consoles, computers, load interface modules may be wire links.

Additional control system elements, such as a limited-function technician's console 24C or a remote focusing device 608, can be connected to the second, third, or other hub unit co-located with the stage or performing area. Other hub units may be connected to additional consoles not co-located with other consoles.

Method of Operation

A method for operating multiple control consoles simultaneously on-line in a computer-controlled lighting

system according to the present invention must resolve potential conflicts between two or more consoles issuing concurrent commands to the same lamp unit or group of lamp units. In a default mode of operation, each lamp unit in a system can be controlled by any console in the system.

Multiple control consoles can be on-line at the same time to permit simultaneous programming of different groups of lamp units. Lamp units in the system can be selected for manual control by multiple consoles simultaneously.

Alternatively, lamp units can be selected for manual control exclusively by only one console at a time. A user-selectable option determines whether lamp units selected for manual control can be seized by another console so that the lamp unit is de-selected at the first console and selected for manual control by a second console, or whether the second console must wait until the lamp unit is de-selected by the operator of the first console before it can be selected for manual control by the second console.

Lamp units in the system can be directed to store and recall cues by any console or computer in the system. Alternatively, cue store and recall operations can be

restricted so that certain lamp units are exclusively assigned to certain consoles for control.

In a second "lamp-unit-locked" mode of operation, any console can issue a command that renders a particular lamp unit or group of lamp units responsive only to that console. In this second mode of operation, other consoles are unable to affect the lamp unit or group of lamp units in any way. The lamp-unit-locked mode can be used to partition the lighting system into groups of lamp units which are controlled by certain consoles and not others. Within each group, lamp units respond to cue store and recall commands from only the one console to which they are "locked."

In a third "lamp-function-locked" mode of operation, any console can issue a command that renders certain functions of a lamp unit or group of lamp units, such as pan and tilt control, responsive only to that console while other functions of the lamp unit(s), such as beam color and intensity, can still be controlled by any other console.

In a fourth "lamp-unit-limited" mode of operation, a control console can be limited to control only certain groups of lamp units, but the lamp units can be controlled by other consoles as well. A user-selectable option

determines whether lamp units currently selected by manual control by the limited console can be seized by another console, or whether the other console must wait until the lamp units are de-selected at the limited console. A user-selectable option determines whether lamp units that are active in a cue recalled by the limited console will respond to cues recalled by other consoles, or whether the lamp units ignore cue recall commands from other consoles if they are currently active in a cue recalled by the limited console.

In a fifth "lamp-function-limited" mode of operation, a control console can be limited to control only certain functions of lamp units, but those functions can be controlled by other consoles as well. A user-selectable option determines whether lamp functions which are selected for manual control or are controlled by an active cue recalled by the function-limited console can be seized by another console or whether the lamp units ignore commands from the other console if those functions are currently controlled by the function-limited console.

A board-level cue or similar macro-function can quickly set up a particular control arrangement. A board-

level cue operable on a single control console, or virtual control console running on a general purpose computer, is a data set that represents a set of known or desired states of various front-panel controls and indicators on that console's front panel. An operator can recall a board-level cue to quickly set the console front-panel features to a desired known state, rather than individually manipulating a large number of front-panel controls to set them to the desired state. Board-level cue recall can take the form of a two or three character board cue identifier which the operator enters on a keypad followed by the press of a "recall" button, thereby reducing board-level set-up to three or four button presses as opposed to 16 to 20 or more button presses which might be required to load cue numbers or chase numbers into several submasters (three-digit cue number and a "go" button, or three-digit chase number and a "load" button for four or five submasters).

A board-level cue operating across multiple control consoles simultaneously on-line can include various modes of lamp-unit-locking and lamp-function-locking, lamp-unit-limiting and lamp-function-limiting for different lamp units or groups of lamp units.

Multiple Console Interaction with Show File

In a distributed control system according to the present invention, a method for operating multiple control consoles simultaneously on-line with one or more general purpose computers must resolve conflicts between two or more consoles issuing commands that affect the show file data base maintained on the system data repository. In systems that include a second or further general purpose computer(s), one or more show file data editing programs may be running simultaneously on the computer(s). There may also be occasions when multiple control consoles and multiple computers running show file data editor programs are on-line simultaneously.

In a default mode of operation, the currently open show file connected to the control resources network can be edited by any control console or show file data editor (show editor) program in the system.

In a second "show-file-locked" mode of operation, any console or show editor can issue a command that renders the show file editable by only that console or show editor.

In this second mode of operation, only one console can

store cues, or only one show editor can add, delete, or modify data in the show file.

In a third "record-locked" mode of operation, any console or show editor can issue a command that renders certain records or groups of records, such as all lamp unit records in a certain cue or all lamp unit records of a certain lamp unit or some other grouping, editable by only that console or show editor. In this third mode of operation, any console or show editor in the system can operate upon all data in the show file except those records which are locked to a particular console or show editor.

Updating the Show File

As cues are stored in the lamp units, a continuously operating background routine updates the show file in accordance with changes to the distributed data base stored in the lamp units.

A show file can be edited directly by a show editor program running on a general purpose computer. As records in the show file are edited, the background routine updates the distributed data base stored in the lamp units in accordance with changes to the show file.

Different Types of Control Consoles

A variety of control consoles optimized for different functions can be implemented by selecting the appropriate mode of operation and user-selectable options as described above for each of a plurality of control consoles having different control devices included on their respective control panels. A technician's console 24C may be function-limited to allow a technician-operator to start and douse lamps, and make small variations in intensity and pan control, but be excluded from controlling color, tilt or other functions, or from storing or recalling cues. A remote focusing device 608 can also be function-limited to allow a lighting director to sequence through active lamp units in reference cues or direct cues and adjust their focus (pan, tilt, beam-size, and edge/optical focus).

A master console operates in the default mode having control over all lamp units in the system not partitioned-off in lamp-unit-locked mode to a secondary console. A secondary console can operate in the lamp-unit-locked mode to control a subset of the entire lighting system, such as work lights, house lights, or other lamp units not used for illuminating the main stage or performance area, while a

master console controls main stage illumination. By partitioning-off peripheral lamp units to a secondary console, the operator of the main console can select all lamp units for manual control and obtain control of only those lamp units designated for main stage illumination, peripheral lamp units being exempted.

Additionally, a hub unit, such as hub unit 606, can isolate data link traffic between consoles, load interface modules, and computers attached to that hub from data link traffic between control devices attached to other hubs. In this way, a technicians console 24C and general purpose computer 560C can design, edit, and download to appropriate lamp units image files for projection by lamp units equipped with liquid crystal projection gates array such as disclosed in U.S. Patent No. 5,282,121 without introducing excessive and irrelevant data link traffic into other segments of the high-speed control resources network 600.

A designer's remote console is a display-only device from which its user can direct its display of data. The designer's remote console communicates with the system data repository over the high-speed data link network and can display the current parameters of the lamp units in the

system, can display cue data for any cue in the show file, and can display any of the display windows used at the master or primary control console.

Display Concepts

The distributed control system of the present invention possesses the capability to display graphical representations of the lighting system in a variety of formats. Based upon a three-dimensional model of the lighting system, which model is contained in the show file data base maintained on the system data repository, each graphic display format conveys information to an operator representing a current state of the system, or as cues are recalled in the model, a possible state of the system.

Photo-realistic Display

One graphic display format presents a photo-realistic view of a stage, set pieces and performers, and how they look under the current illumination from the lamp units. This view can be used to simulate show designs as a communication or sales tool for lighting designers. This view can also be used to provide realistic feedback for offline show programming and for control locations having an obstructed view of a stage. In the photo-realistic view,

lamp units are represented as projecting full conical or columnar beams which are visible only to the extent at which smoke is considered to be present in the air. A pool of light projected by a lamp unit, or the pattern of its gobo, is mapped onto surfaces illuminated by the lamp unit. Various surface materials are employed in the model, taking into account the reaction to illumination and reflective properties of the surface materials. Video screens may be included in the model with an operator being able to play real video sequences on the modelled screens. As cues are recalled in the model, the photo-realistic view shows a smoothly animated transition of beams in the air and of projections as they move. The operator's point-of-view can be quickly and easily moved to allow viewing the model from any angle. The operator can move to preset points-of-view with a single button press.

Working-view Display

Another graphic display format presents a working view of the model comprising a mix of less realistic three-dimensional graphics, two-dimensional graphics, and numeric data. Three-dimensional graphics provide a visual context for other data. Two-dimensional graphics present lamp unit

information in a visual and qualitative way. Numeric data presents lamp unit information in a quantitative way. The stage, set pieces and performers are presented less realistically, in a more block-like way. No embedded video screens are included, nor surface modelling, nor modelling of the lamp units' beams and projections. Light beams projected by active lamp units are represented by colored bars. Graphical effects on the colored bars represent pattern projections (gobos), hard or soft edge effects (optical focus) and "marked" lamp units (units which conform to all variable parameters but with no light beams actually projected in a given cue). A beam terminus can be represented by a colored dot or other graphical device to indicate the target at which the lamp unit is pointing. Clicking on a beam bar with a mouse or other pointing device selects the corresponding lamp unit for manual control. The beam terminus can be dragged by the pointing device to re-focus lamp units. The terminus snaps to the target surface or the head of a performer or to other default points to make aiming the beam easier. Once a geometric pattern of beams has been arranged, the beams can be grouped together and thereafter moved as a group by dragging any one of the

beams. Clicking on a group of beams selects the corresponding group of lamp units for manual control.

When cues are recalled, the working view shows a rough animated transition of beam bars in the air as they move. The operator's point-of-view can be quickly and easily moved to allow viewing the model from any angle. The operator can move to preset points-of-view with a single button press.

The operator can have a fish-eye view from just off the front edge of the stage, halfway between the stage and the overhead lighting truss, giving a view of the truss from below and the stage from above. The motion of the operator's point-of-view is constrained to circle around the edge of the stage in the fish-eye working view.

The working view can superimpose numeric display of a single, selected parameter over the body of each lamp unit. The operator can switch on or off groups of lamp unit beam bars and numeric displays; lamp units can be displayed in "pages"; groups of (typically 100) contiguous lamp unit addresses. A "magic wand" selection tool is provided so that when the operator clicks on a beam to select the corresponding lamp unit for manual control, all active lamp

units having the same beam color as the selected lamp unit are also selected for manual control. When the operator clicks on a target using the magic wand tool, all active lamp units focused on that target are selected for manual control. Lamp units selected for manual control are also selected for numeric display, allowing the operator to view numeric data for only some of the lamp units on a page.

Channel-usage Display

Another graphic display format presents a channel usage display which, while maintaining the three-dimensional model of the lighting system, stage, set pieces and performers, presents a view that breaks the model. Intended as a data visualization tool rather than a realistic view of the stage, the channel usage display can be manipulated to show all of the important surfaces of the stage, set pieces and backdrop in one view rather than having to move the operator's point-of-view to see parts of the set that are hidden by other parts of the set. The operator's point-of-view is fixed in the channel usage display, but portions of the model can be moved and rotated in the view to bring their important surfaces into view from the fixed point-of-view. For example, set pieces can be viewed from in front

of the stage, the stage can be tilted down so that its top surface is seen, and a backdrop can be raised from behind the set pieces so its lower portion can be seen. Lamp units themselves are not shown, nor are beams in the air. A line marking the outline of a beam projected on a set piece or other surface is drawn; some graying effect may be used to fill the outline of a pool of light formed by the beam projected on the surface. A lamp unit address or control channel number can be displayed in the center of the beam's projected area, or can be turned off if multiple beams converge on a single point. Lamp units can be selected for manual control by clicking on the corresponding beam's projected area, address or channel number, or target using a mouse or other pointing device

Console Features

Control consoles utilized in the distributed control system of the present invention may contain control devices optimized for a variety of functions including but not limited to: 1) selecting and de-selecting lamp units or groups of lamp units for manual control, 2) adjusting variable parameters of lamp units selected for manual control, 3) storing variable-parameter data describing the

present state of lamp units in the system for later recall, 4) recalling and conforming variable-parameter lamp units to stored variable-parameter data, 5) uploading stored variable parameter data from lamp units to a system data repository, 6) downloading stored variable parameter data from a system data repository to lamp units, and 7) requesting and displaying status information related to the present state of lamp units in the system.

Whereas some control consoles, such the Artisan® and mini-Artisan® control consoles manufactured by Vari-Lite, Inc. of Dallas, Texas, have used separate front-panel features for recalling single stored sets of variable parameter data (direct cues) and automatically-executable sequences of stored sets of variable-parameter data (chases), control consoles according to the present invention utilize flexible front-panel features capable of recalling either a direct cue, a chase sequence, or other playback function.

A front-panel feature called a "submaster" is normally adapted either for recalling direct cues, for recalling and executing chases (sequences of direct cues), controlling cross-fade between two direct cues, or

controlling intensity of groups of lamp units distributed over a group of matrix faders; a number of direct cue submasters and a number of chase sequence submasters being normally provided on a console front panel. A submaster is normally provided with a variable controller or "fader" used to control the overall intensity level of lamp units assigned to the submaster, a "go" button for loading a direct cue number to the submaster or a "load" button for loading a chase sequence identifier to the submaster, and a pair of buttons for incrementing or decrementing the direct cue number loaded to the submaster or a pair of buttons for running and stopping or single-stepping through cues in a chase sequence. A submaster "select" button is provided to activate or de-activate cues or chases recalled to the submaster, so that the cue or chase can be loaded prior to execution and then executed by a single button press at the appropriate moment.

Consoles according to the present invention utilize soft-submasters, such as those shown by way of example in figure 33, which can respond either as direct-cue submasters, chase-sequence submasters, or other submasters depending upon how a direct cue number, chase sequence

identifier or other identifier is loaded to the submaster. If the operator enters a cue number on a numeric keypad and presses a "direct" button associated with the keypad, then presses a "go" button associated with a soft-submaster, the submaster automatically becomes a direct-cue submaster and the other buttons associated with the submaster assume states in which they perform appropriate functions such as to increment or decrement the cue number that is recalled to the submaster. If the operator enters an chase-sequence identifier on an numeric keypad and press the "chase" button associated with the keypad, then presses the "go" button associated with a soft-submaster, the submaster automatically becomes a chase-sequence submaster and the other buttons associated with the submaster assume states in which they perform appropriate functions such as to run or stop the chase sequence or to single-step through the sequence of cues.

Groups of soft-submasters can be configured to perform other playback functions such as cross-fading between two direct cues and individual control of intensity of groups of lamp units in a single direct cue (matrix control) by entering a number on a numeric keypad and

pressing button identifying the a particular playback function. For example, an operator can enter a cue number at the keypad, press an "X-fade" button associated with the keypad, and press "go" at a soft-submaster, then enter another cue number, press "X-fade" and press "go" at another soft-submaster to configures the pair of submasters as a cross-fade controller that fades between the two direct cues.

A similar technique may be employed to designate matrix faders and a matrix submaster. A matrix submaster controls the overall intensity of groups of lamp units, the overall intensity of each group being controlled by one of group of matrix faders assoicated with other soft-submasters.

Consoles according the present invention utilize programmable display elements to change the labels on the buttons associated with soft-submasters depending upon the required functions. These may take the form of buttons mounted adjacent to separate display elements, or the display elements may be incorporated in the button or keycap itself.

Whereas a typical submaster has incorporated features for enabling or disabling timing effects stored with the cues and for filtering-out certain of the variable parameter functions of lamp units that are active in cues recalled to the submaster, consoles according to the present invention may also include features for filtering-out certain lamp units, on a channel-by-channel basis, that would otherwise be active in the cue.

Cue Structure

Control consoles according to the present invention are provided with front-panel features and operating system programs that enable the use of a multi-level cue structure that includes board-level cues, direct cues, and reference cues. A direct cue is a data set that may contain variable-parameter data in the form of absolute parameter data for each function of a lamp unit, but a direct cue may also contain a reference identifier in the place of any one or more absolute values for any function, the reference identifier serving as an index that points to another data set that contains the absolute parameter data.

A reference cue is another data set that contains absolute parameter data that may be referred to by a direct

cue so that all direct cues built from a given reference cue can be updated to new absolute parameter data simply by modifying the data in the reference cue.

A board-level cue is a data set that describes the state of various front-panel features of a control console at a given moment, and includes, for example, which direct-cue numbers, chase-sequence identifiers, cross-fade cues, and matrix patch configurations are loaded into which soft-submasters at that moment. Front-panel features for storing and recalling board-level cues are provided separately from features for storing and recalling direct cues, chase sequences and the like for the soft-submasters. Front-panel features for storing and recalling reference cues may be provided separately from features for storing and recalling direct cues, or some features such a numeric keypad can be shared by direct cue and reference cue controls; there being provided a button for "store reference cue" separately from a button for "store direct cue."

A board-level cue can be stored with "clear" or "thru" status for individual submasters. "Clear" status removes any previous submaster entry from a submaster when the board-level cue is recalled. "Thru" status leaves any

previous submaster entry in a submaster when the board-level cue is recalled. Board-level cues can trigger and be triggered by external events by transmitting board-level cue operations over the control resources network, listening at each console for board-level cue activity on the network, and interpreting board-level cue activity depending upon the prior state of the console. A console can then be programmed to respond to board-level cue activity on another console so that multiple control consoles on a network can be configured by one board-level cue recall at one of the consoles on the network.

Tracking

A tracking function can be applied to a numerical sequence of direct cues having identical parameter values for one or more lamp unit. For example, cues 10.01 through 10.09 may all specify the same color for one or more lamp units while specifying different pan and tilt values in each cue. In this example and according to the tracking feature of the present invention, the first cue of the sequence (10.01) is a "source cue" containing the parameter data to be tracked, and the color function is made to track through the numerical sequence of cues so that if at any time the

color is changed for any of the cues, the color is thereby changed in all cues of the sequence subject to the tracking feature. A user-selectable option determines whether changes made to the tracked function apply only from the current cue forward to the end of the sequence (thereby establishing a new source cue), or whether a change made in one cue of the sequence is applied throughout the sequence from the original source cue (10.01) to the end (10.09). A source cue can also contain the identifier of a reference cue instead of actual parameter data.

Tracking can be applied in one of two modes. In a standard mode, tracking functions through all consecutive sequential cues in which a particular lamp unit parameter remains unchanged throughout the sequence, but stops tracking when that parameter changes, even if it changes back to the original parameter value in subsequent cues of the sequence. In an optional mode, tracking resumes for subsequent cues of the sequence in which the parameter value returns to the original value. A user-selectable option may be utilized to determine whether the standard or the optional mode applies.

Tracking can also be applied to a chase, with the first cue of the chase sequence being a "source cue" for tracking data, and all subsequent cues of the chase tracking the source cue; this establishes an arbitrary or random sequence of cues since chases can be programmed as arbitrary or numerically-sequential lists of cue numbers.

Previously, chase sequences executed with a single chase rate value controlling the time between sequential cue recall operations. Control consoles according to the present invention are provided with front-panel features and operating system programs that enable the use of variable timing between each cue of a chase sequence. This can take the form of a first chase rate that applies to sequential execution of cue recall operations from a first cue of a sequence to a subsequent cue for which a new chase rate is specified. A user-selectable option determines whether the new chase rate is established for all subsequent cues (a "standard" mode) or whether the new chase rate applies only to the one cue which the new chase rate precedes (an "exception" mode) after which the first chase rate is re-applied. Multiple new chase rates can be inserted into a single chase sequence. Each new chase rate can be flagged

to operate as a standard-mode change or and exception-mode change.

Timing Control

Direct cues can include timing control, speed control, and/or wait times for individual functions of lamp units that are active in the cue. Timing control causes one or more variable functions to transition from a prior state to a currently commanded state over a specified time; separate times being possible for different functions such as pan, tilt, color, beam-size, edge, gobo- or image-rotation, intensity and the like. Speed control causes one or more variable functions to transition from a prior to state to a currently commanded state at a specified rate of change; separate speeds being possible for different functions. Wait times cause one or more variable functions to delay the start of transition from a prior state to a currently commanded state; separate wait time being possible for different functions. A direct cue can have multiple wait times for various functions coupled with timing control or speed control for those functions such that, following the programmed wait time, the transition from a prior state

to a currently commanded state is then modulated by the programmed timing or speed control.

Timing control data that can be stored with each cue may include a simple "one time for all parameters," or may include separate times for each of a few broad classes of parameters such as intensity, focus, color, and beam parameters. Control consoles according to the present invention further include front-panel features, such those shown by way of example in figure 34, and operating system programs that enable the programming of separate timing control values for each variable parameter function of a lamp unit. For example, where the color-control function is controlled in a particular lamp unit by three motorized color filter drives, the timing control data stored with a particular cue for that lamp unit may include separate execution times or speed control values for each of the three color filter drives. Alternatively, a single timing control value can apply to all three color filter drives equally. A hierarchal structure can be imposed on the timing control values so that a single time for all functions, whose default value is zero-time or full-speed, is initially entered for each lamp unit function.

Thereafter, entering a timing value for a broad class function such as color or focus overwrites the single timing value for all specific functions included in the broad class. For example, a color time imposed over a single time for all functions causes new timing values to be entered for each color filter drive of the affected lamp unit.

Subsequently, color timing can be modified by entering a third timing value for only one of the color filter drives of the lamp unit, the other color filter drive(s) being controlled by the timing value entered for the broad class function "color." A user-selectable option determines whether a new timing value entered for a broad class function overwrites all timing values for specific parameter drives (a "reset" mode) or whether the new timing value is only propagated to parameter drives that have not been programmed with specific timing values (a "fill-in" mode).

Wait times can be entered for each of the broad class functions of a lamp unit in a cue, or can be entered for each of the specific parameter drives for each broad class function. Wait times can be programmed according to the same hierarchal structure described above with respect

to timing values, and can be implemented with "reset" or "fill-in" modes as described above.

An auto-follow mode can be applied to board-level cues so that the board-level cue initiates automatically following the completion of a previous event. In this way, one board-level cue can automatically follow another to simplify the execution of a complex sequence of events. The auto-follow mode can also be applied to direct cues so that numerically sequential direct cues are recalled automatically, observing any wait times and timing control values stored with the cues, so that the autofollow will always occur regardless of how the initial cue was loaded.

Intelligent Lighting Assistant

The distributed control system of the present invention includes control system elements and operating system programs that enable the system to respond to a human operator as an "intelligent lighting assistant." A reasoning engine implemented on a general purpose computer connected to the control resources network accepts command inputs from a voice-recognition module or other input device. Voice recognition devices and techniques are well-known to those with skill in the art; for example the

ProAudio 16 sound card and ExecuVoice software available from Media Vision of Fremont, California.

The reasoning engine evaluates the command by referring to the show file, which contains a three-dimensional model of the lighting system and a data base of parameter data sets (cue data) for lamp units modelled therein. The show file is maintained on a system data repository also connected to the control resources network.

The reasoning engine operates utilizing a method illustrated in figure 35 for translating generalized instruction inputs into parameter adjustment data outputs. The method includes applying a set of rules for interpreting generalized and subjective command inputs, such as the phrase "the upstage lamp units in red change to blue," into lamp unit address to select for manual control and parameter adjustment values to drive the lamp units to produce light beams having the required attributes or parameters. The reasoning engine, according to the method of the present invention, analyzes the command input to determine which lamp units will be affected by the command, and also determines what the effect of the command will be upon those lamp units. Finally, the console composes appropriate

system commands and transmits the commands over the high-speed control resources network 600.

A user-selectable option determines whether a separate "go" command input is required before the affected lamp units respond to the command input, or whether the resultant adjustments are performed immediately upon interpretation of the command by the control system.

The reasoning engine according to the present invention can distinguish common-language expressions describing relative locations of objects contained in the three-dimensional model maintained on the system data repository, their support structure hierarchy (which objects are supported by other objects), attributes of the various objects (size, shape, color, type, names, control channel numbers or other designations), and the various possible states of parameter drives in variable parameter lamp units. Words, phrases, and other expressions having specialized meanings in the lighting industry are interpreted by the reasoning engine so that the resultant parameter adjustment values at least tend to approximate the general understanding in the industry of specialized meaning. Other words, phrases or expressions are given specialized meanings

as required to avoid unpredictable results in cases where an expression may have two or more meanings. The reasoning engine of the present invention, therefore, correctly interprets industry "jargon" and may expand the jargon to include more expressions having specialized meanings.

Knowledge used for evaluating the input to determine channel selection requirements includes common knowledge terms such as "first," "last," "odd," "even" and the like; technical jargon such as "first electric," "torm" and the like; knowledge derived from the three-dimensional model such as "upstage," "stage right," "back truss" and the like; and current lamp unit attributes such as "green lights," "lights on the singer" and the like. Knowledge used for evaluating the input to determine parameter adjustment requirements includes console structures such as "preset color," "preset focus" and the like; numeric parameter entry such as "at intensity 75," "in gobo 3" and the like; knowledge derived from the three-dimensional model such as "at the singer," "two feet upstage" and the like; and current lamp unit attributes such as "the same green as those other lamp units" and the like.

The reasoning engine of the present invention interacts with the system data repository and graphic displays associated with the control resources network to display the results of command input expressions in both on-line and off-line situations. An operator can program a lighting system using only the intelligent lighting assistant, comprising a command input interface such as a voice-recognition module, a reasoning engine implemented on a general purpose computer, and a three-dimensional model of the lighting system. The model simulates the activity of lamp units in the absence of the rest of the lighting system, while the command input interface and reasoning engine substitute for control consoles to provide for off-line programming. Results of command input expressions are displayed on a graphic display showing a photo-realistic view, a working view, or a channel usage display as described above. Alternatively, multiple graphic displays can be utilized showing different views on different displays. A voice-synthesizer module can be included in the system to allow the "intelligent lighting assistant" to speak to the operator.

In case of ambiguities which the reasoning engine cannot resolve, a question can be posed to the operator, either as text on a graphic display or as an electronically synthesized voice. For example, to evaluate an input containing the expression "lights on the singer," the system may pose the question "Which singer do you mean?" This question may be communicated audibly, utilizing a voice synthesizer module, or the question may be displayed as text in a window on the graphic display. As another example, if the input contains the expression "the same green as those other lamp units," the system may pose the question "Which other lamp units do you mean?"

4. Brief Description of the Drawings

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following description taken in conjunction with the accompanying drawings, in which:

FIGURE 1 is a perspective view of a computer-controlled lighting system of the type which may embody the present invention as set up for illuminating a stage;

FIGURE 2 is a block diagram of the lighting system of the type which may embody the present invention, which illustrates the communication between the control console and the various lamp units as well as other items of stage equipment;

FIGURE 3 is an illustration of the front panel for a control console for a lighting system of the type which may embody the present invention;

FIGURE 4 is a block diagram for the electronic subsystems which are of the type which may be a part of the control console;

FIGURE 5 is an electronic block diagram of the lamp processor system portion of a lamp unit;

FIGURE 6 is a block diagram illustrating a lamp unit stepper control system;

FIGURE 7 is a block diagram illustrating an index sensor system for use with the stepper motors in a lamp unit;

FIGURE 8 is a block diagram illustrating servo feedback control of a motor within a lamp unit including rate of movement control and position monitoring;

FIGURE 9 is a detailed schematic diagram for a repeater as shown in FIGURE 2;

FIGURE 10 is a flow diagram illustrating the operation of programs in the control console which includes a main sequencer that steps through a number of sensing, communication and other operational control programs;

FIGURE 11 is a flow diagram of additional programs utilized in the control console for carrying out the operation of the lighting system of the type which may embody the present invention;

FIGURE 12 is a flow diagram illustrating the individual steps carried out in a lamp unit for initializing the lamp unit to begin operation (part 1);

FIGURE 13 is a flow diagram illustrating the individual steps carried out in a lamp unit for initializing the lamp unit to begin operation (part 2);

FIGURE 14 is a flow diagram illustrating the basic operation of programs in the processor of the lamp unit including a main sequencer program which steps through a command reception unit and a series of test programs;

FIGURE 15 is a flow diagram illustrating the operations carried out within the lamp processor for receiving parameter control commands, processing these commands and directing the physical operations that are carried out by mechanisms within

the lamp unit for causing the light beam to have a selected set of parameters;

FIGURE 16 is a block diagram of a repeater as shown in FIGURE 9;

FIGURE 17 is a block diagram showing the interconnection of two signal distribution racks in master/slave relationship;

FIGURE 18 is a block diagram of the lighting system of the type which may embody the present invention, showing the master (console) repeater, slave (trunk) repeaters, and the various truss repeaters connecting the console and the lamp units;

FIGURE 19 is a block diagram of the lighting system showing the existing broadcast network;

FIGURE 20 is a block diagram of the lighting system showing the existing reply network;

FIGURE 21 is a block diagram of an improved repeater of the type which may embody the present invention;

FIGURE 22 is a block diagram of another improved repeater of the type which may embody the present invention;

FIGURE 23 is a block diagram of a "smart" repeater of the type which may embody the present invention;

FIGURE 24 is a block diagram of the lighting system showing an improved broadcast network of the type which may embody the present invention;

FIGURE 25 is a block diagram of the lighting system showing an improved reply network of the type which may embody the present invention;

FIGURE 26 is a block diagram of the electronic subsystems forming part of a modular controller mainframe;

FIGURE 27 is a block diagram of the modular control console system;

FIGURE 28 is a block diagram depicting a distributed control system;

FIGURE 29 illustrates typical architecture for a control console;

FIGURE 30 illustrates typical architecture for a general purpose computer;

FIGURES 31 and 32 illustrate typical architecture for load interface modules;

FIGURE 33 illustrates a soft-submaster;

FIGURE 34 illustrates various front panel features; and

FIGURE 35 illustrates a method for translating generalized instruction inputs into parameter adjustment data inputs.

[Explanation on Reference Numerals]

20	system
22	stage
24, 82, 84	control console
26	data link
28	lamp unit
30	lamp
32	floor lamp
42, 44, 46, 48, 50	lamp unit
52, 54, 56, 58	repeater
60	dimmer
62, 64	control signal converter
66	chain hoist motor
68	air cannon
70	special effects projector
80	bi-directional bus
178	lamp processor system
190	encoder/decoder
200	microprocessor
202	RAM and EPROM memory
218, 220	decoder
246	multi-protocol communication controller
254	parameter drive circuit
278, 296	parameter control circuit

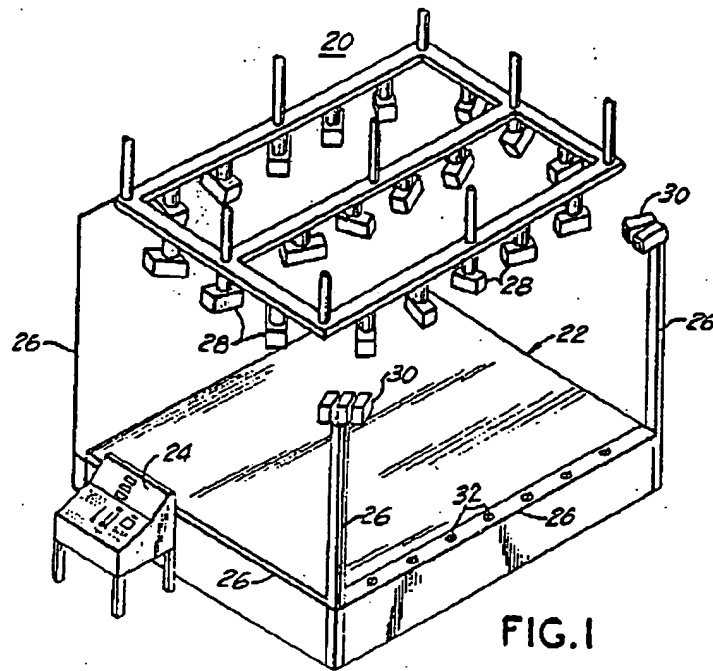


FIG. 1

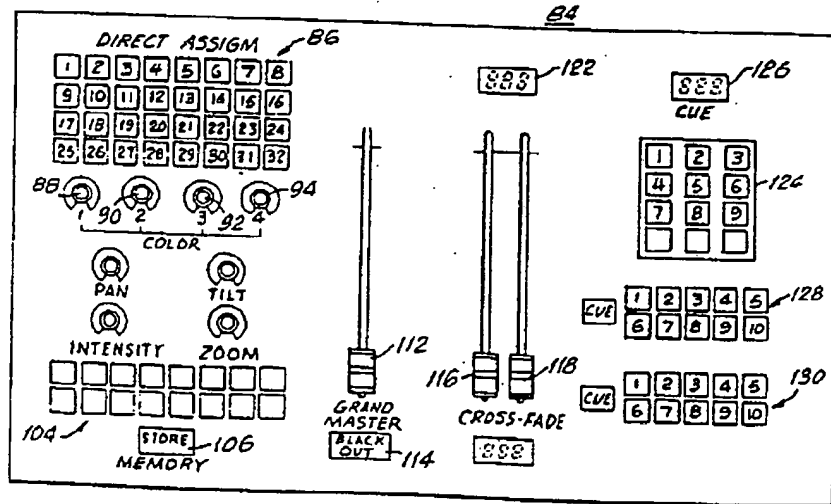


FIG. 3

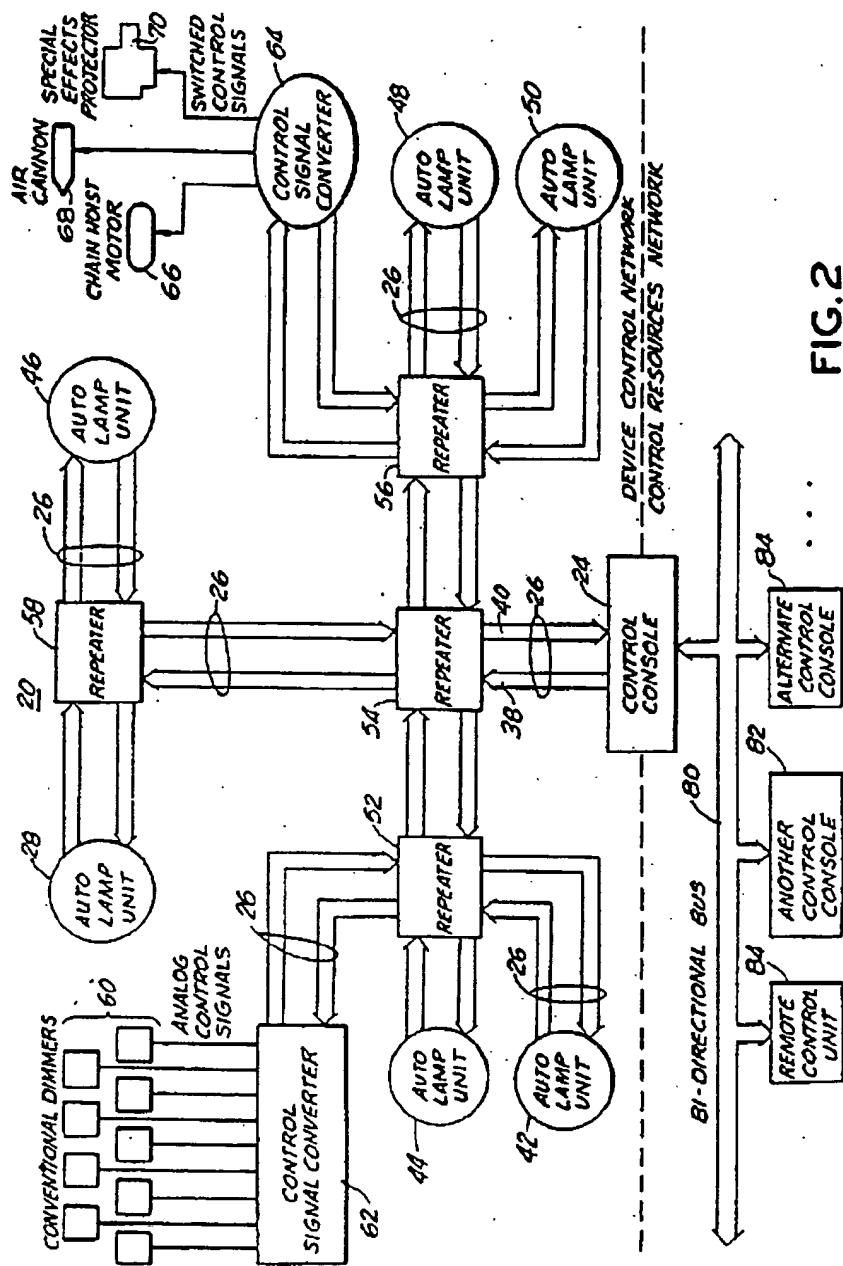


FIG.2

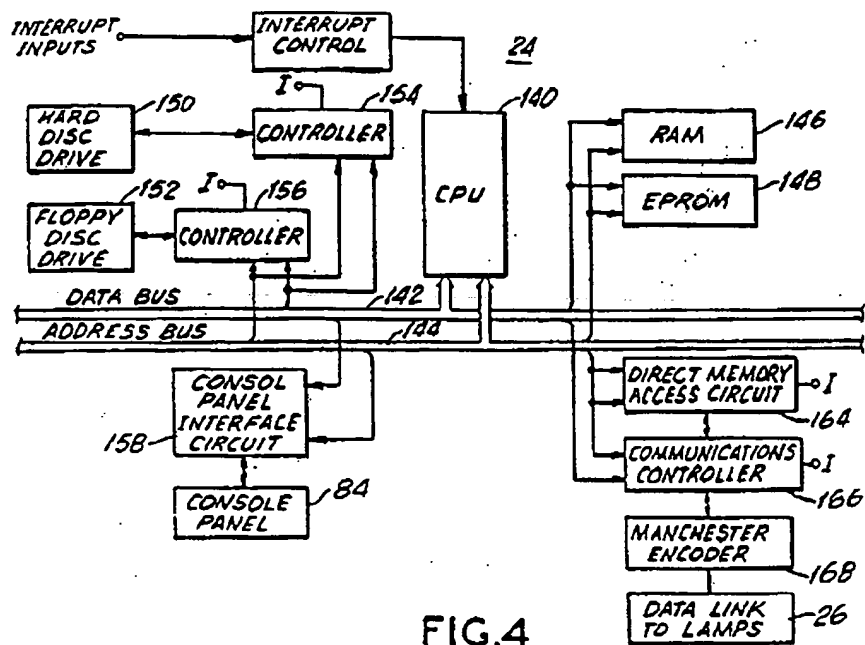


FIG. 4

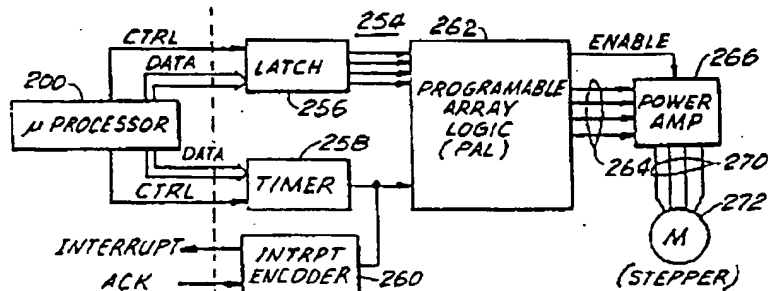


FIG. 6

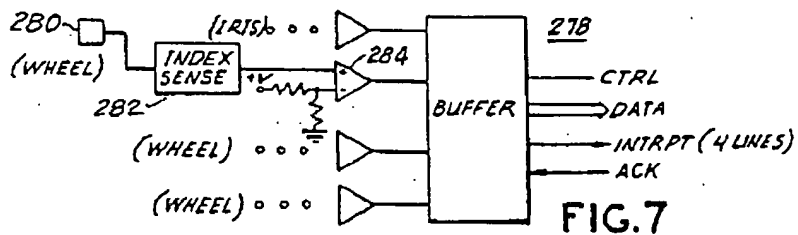


FIG. 7

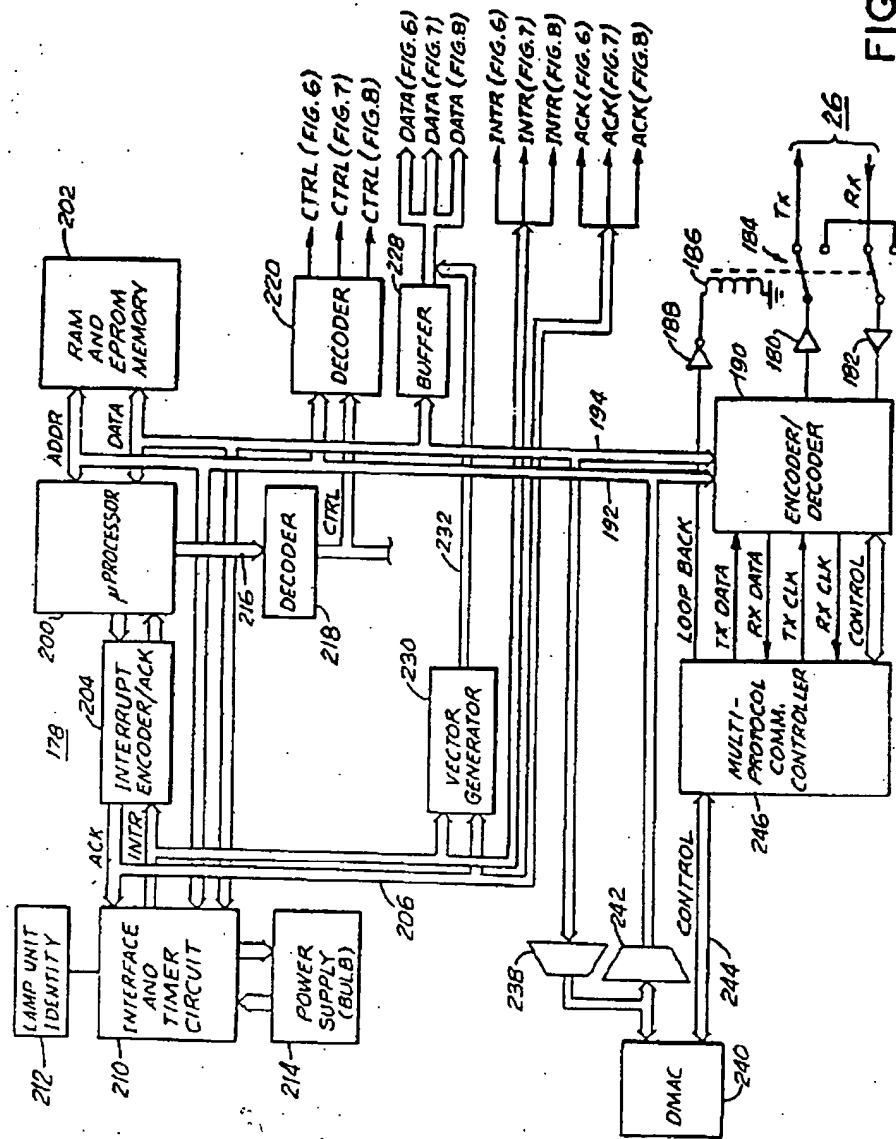


FIG. 5

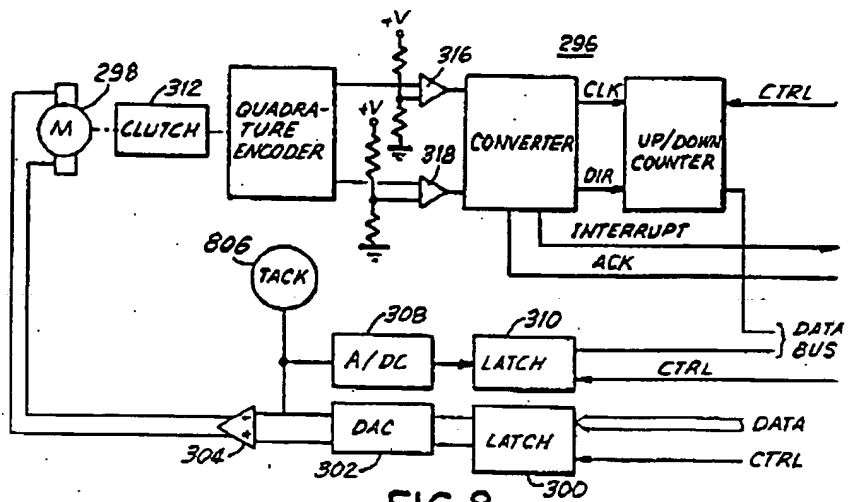


FIG. 8

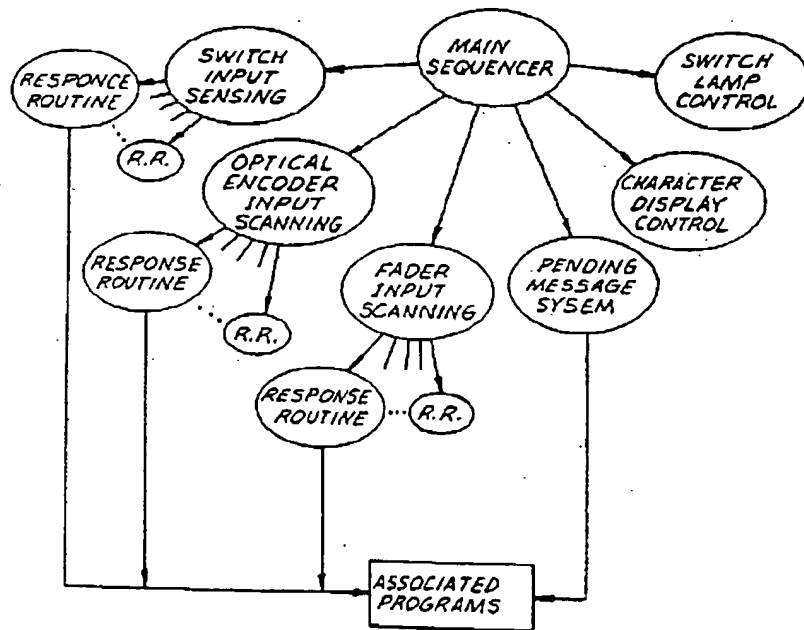
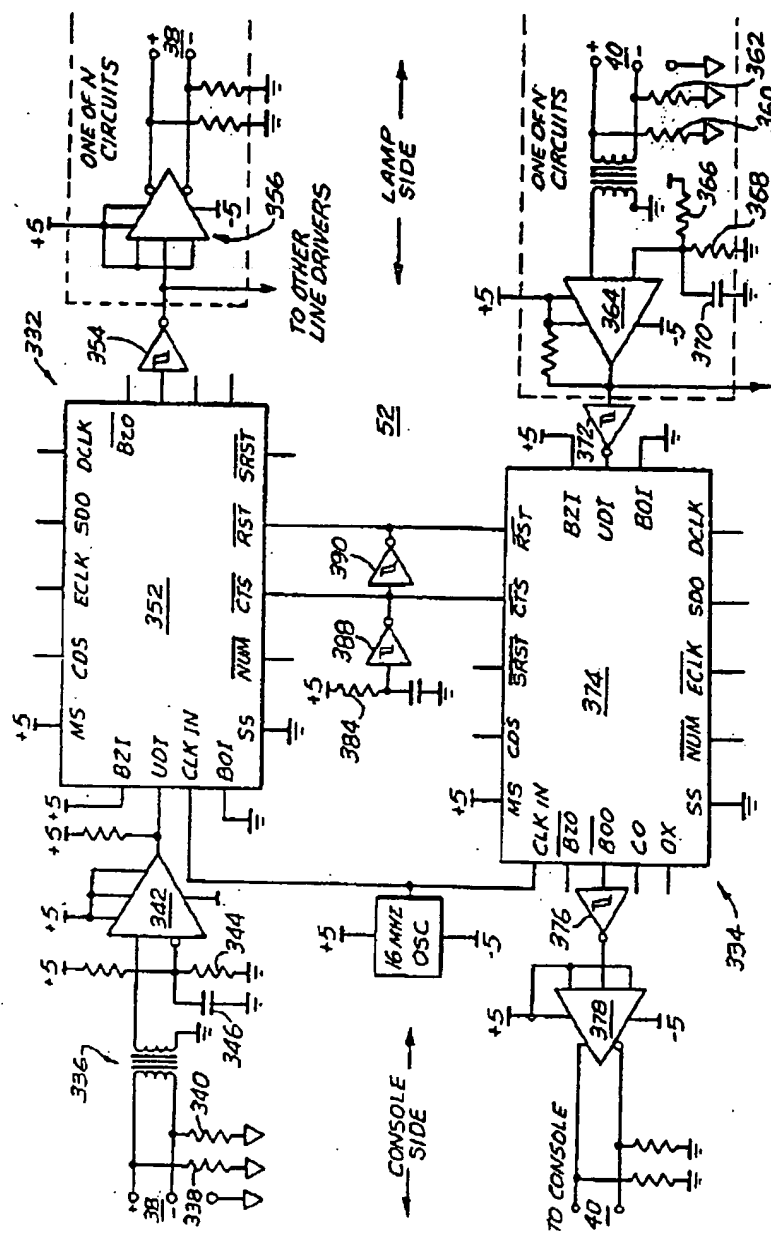
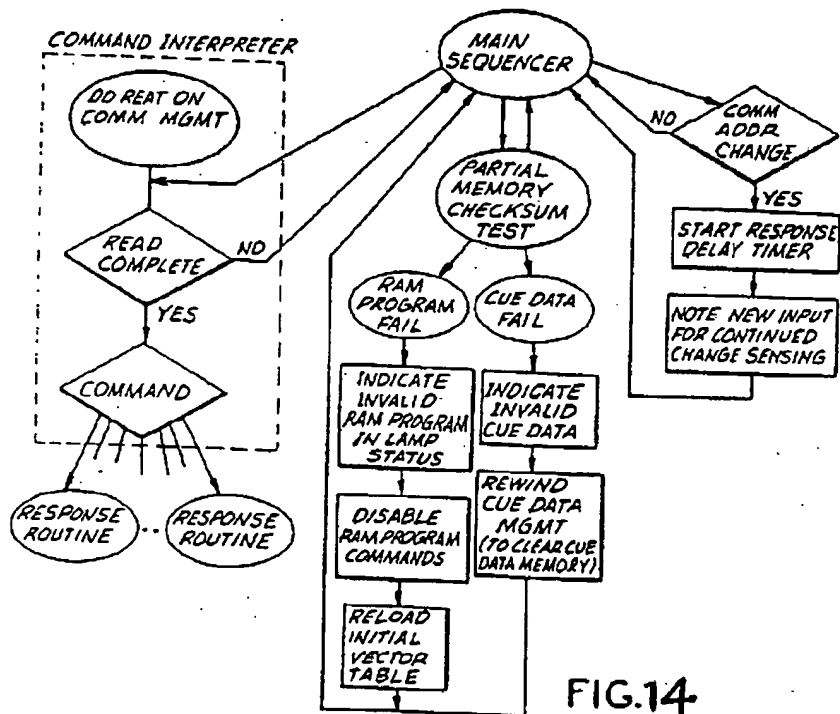
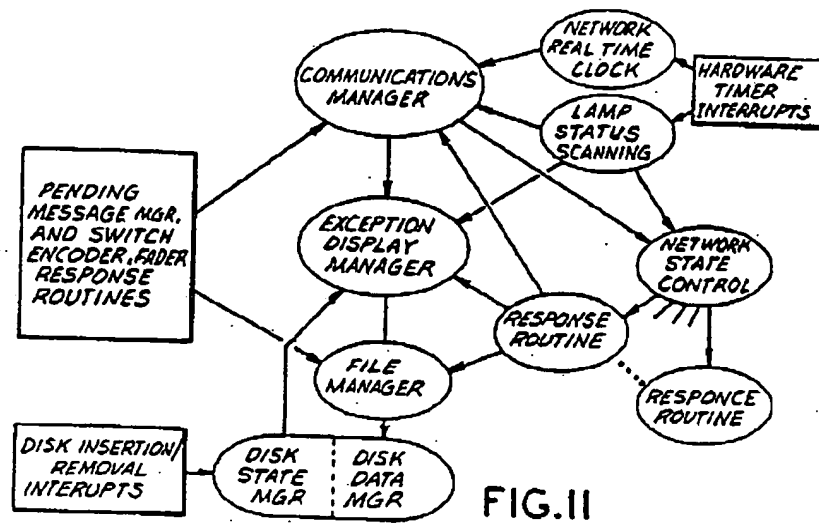


FIG. 10





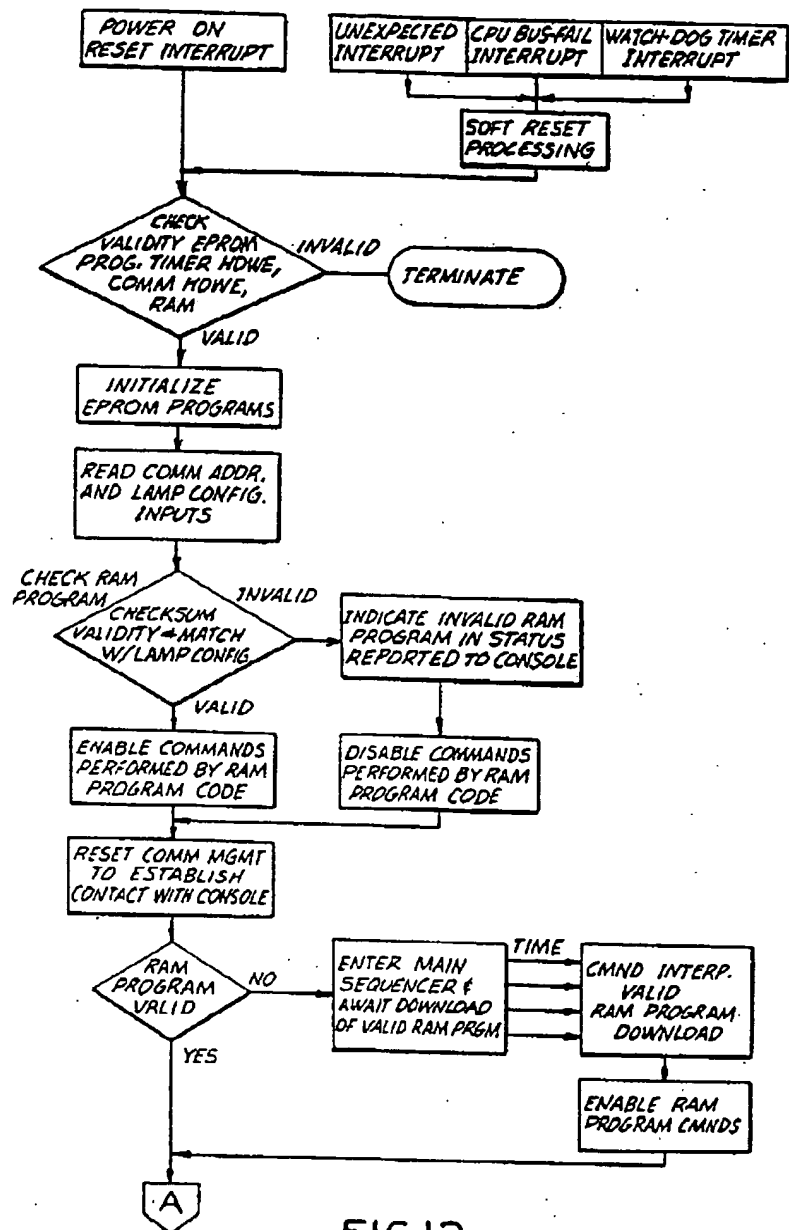


FIG.12

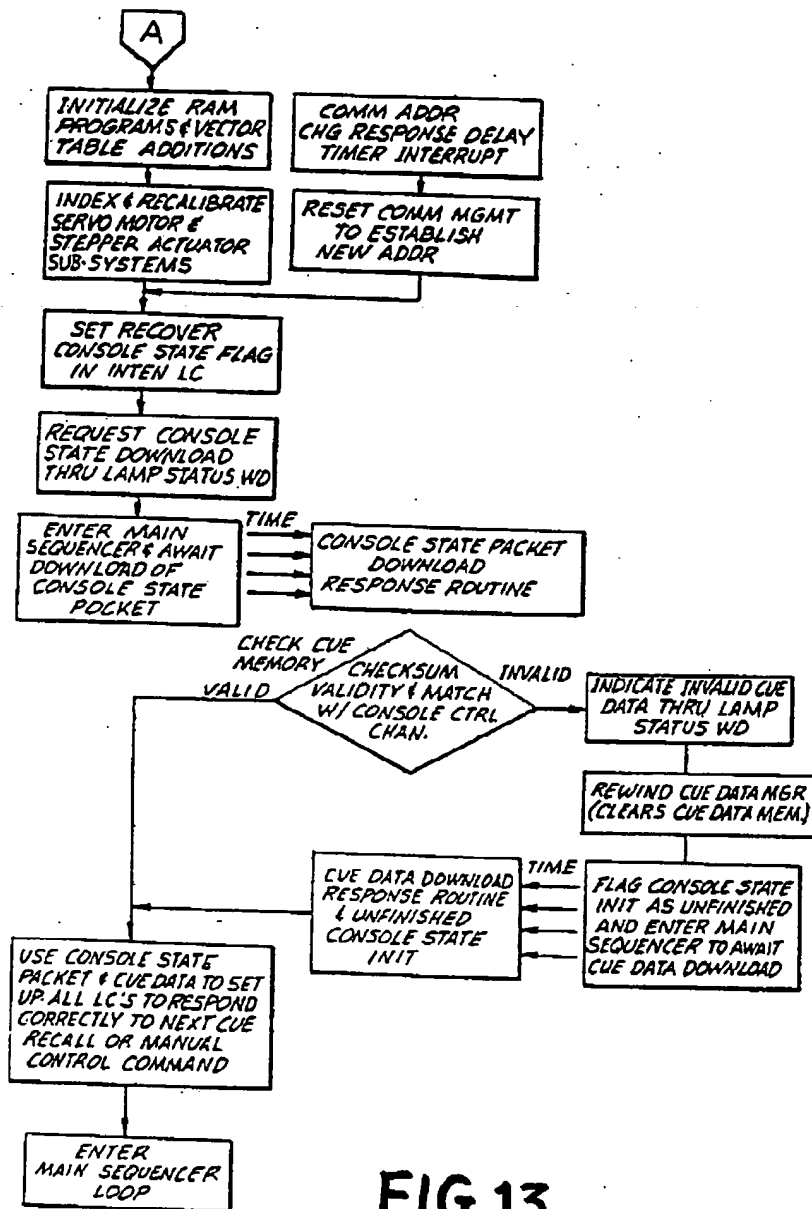


FIG.13

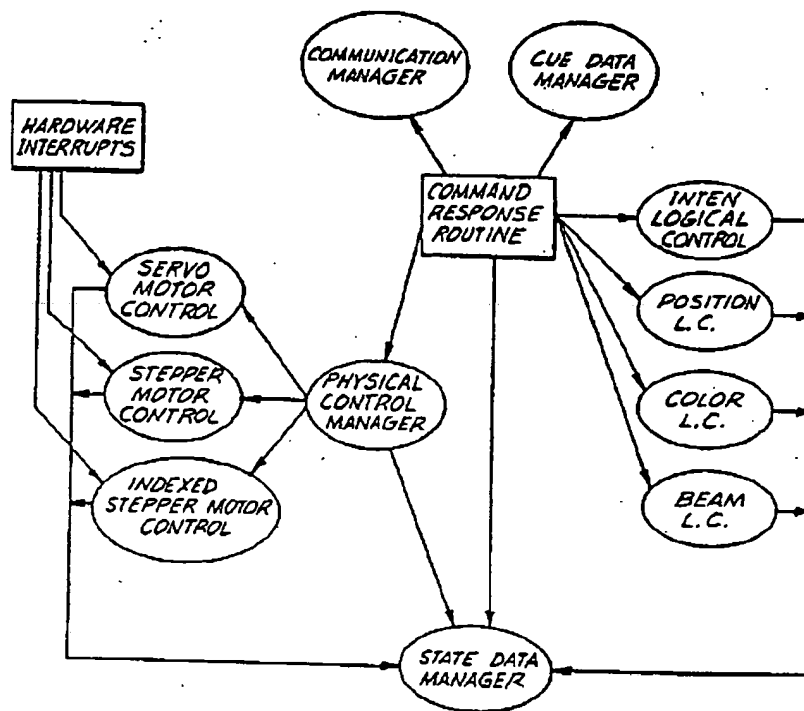


FIG.15

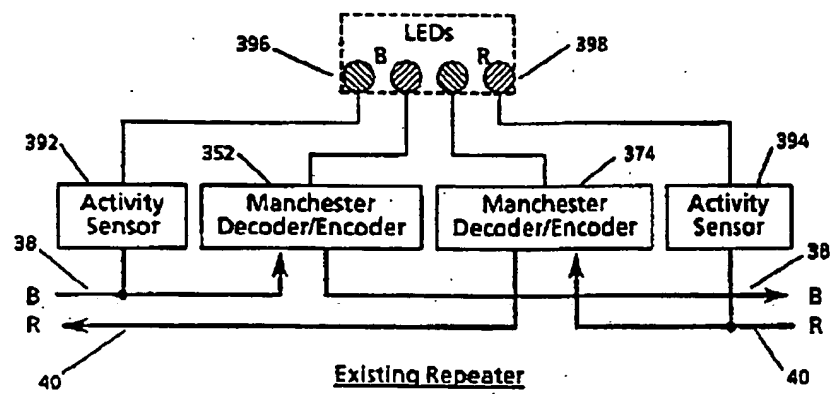


FIG. 16

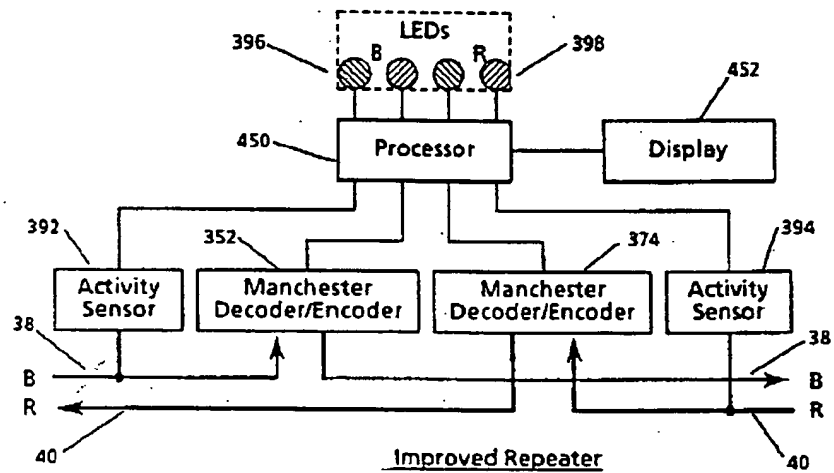


FIG. 21

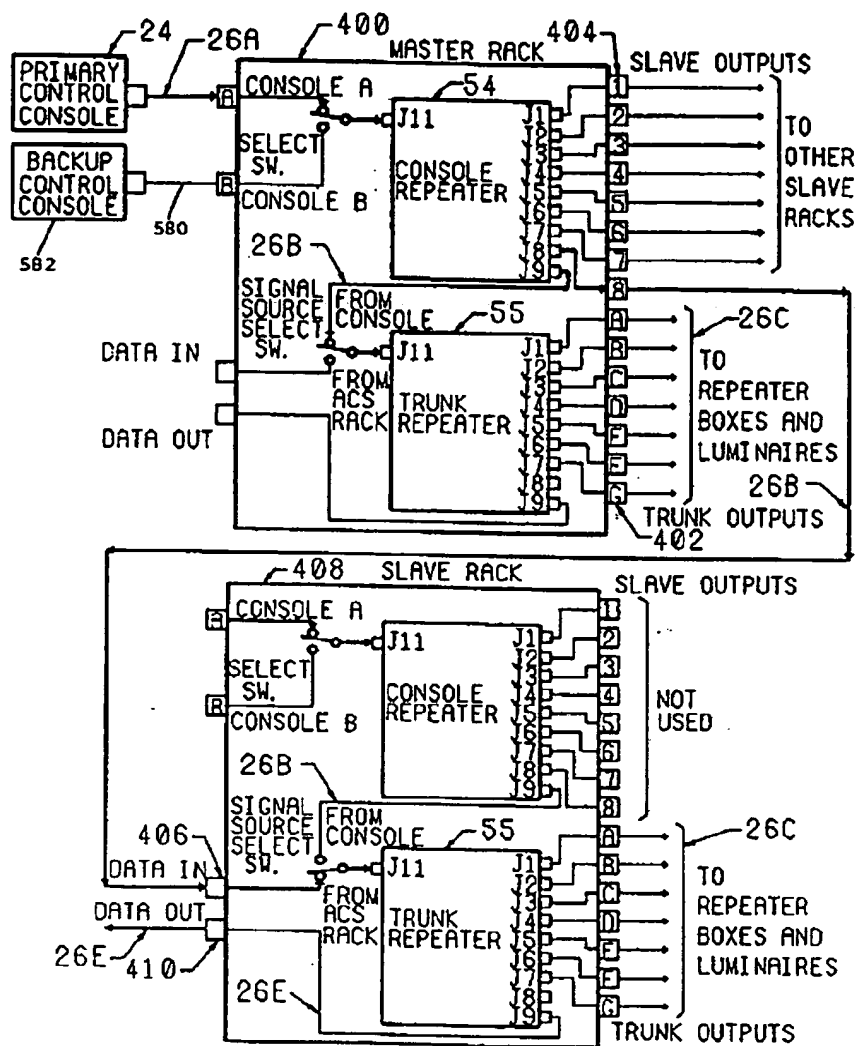


FIG.17

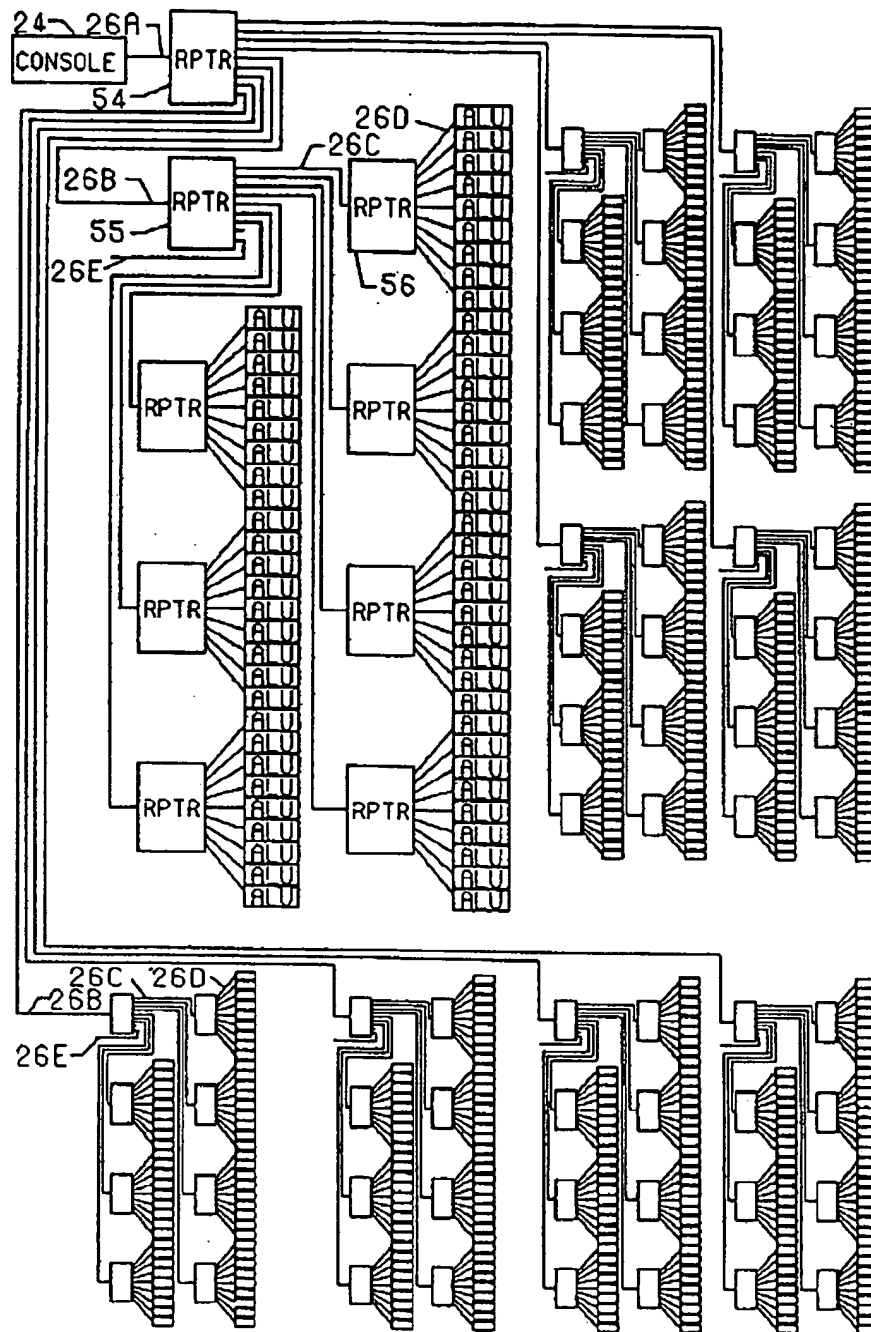


FIG. 18

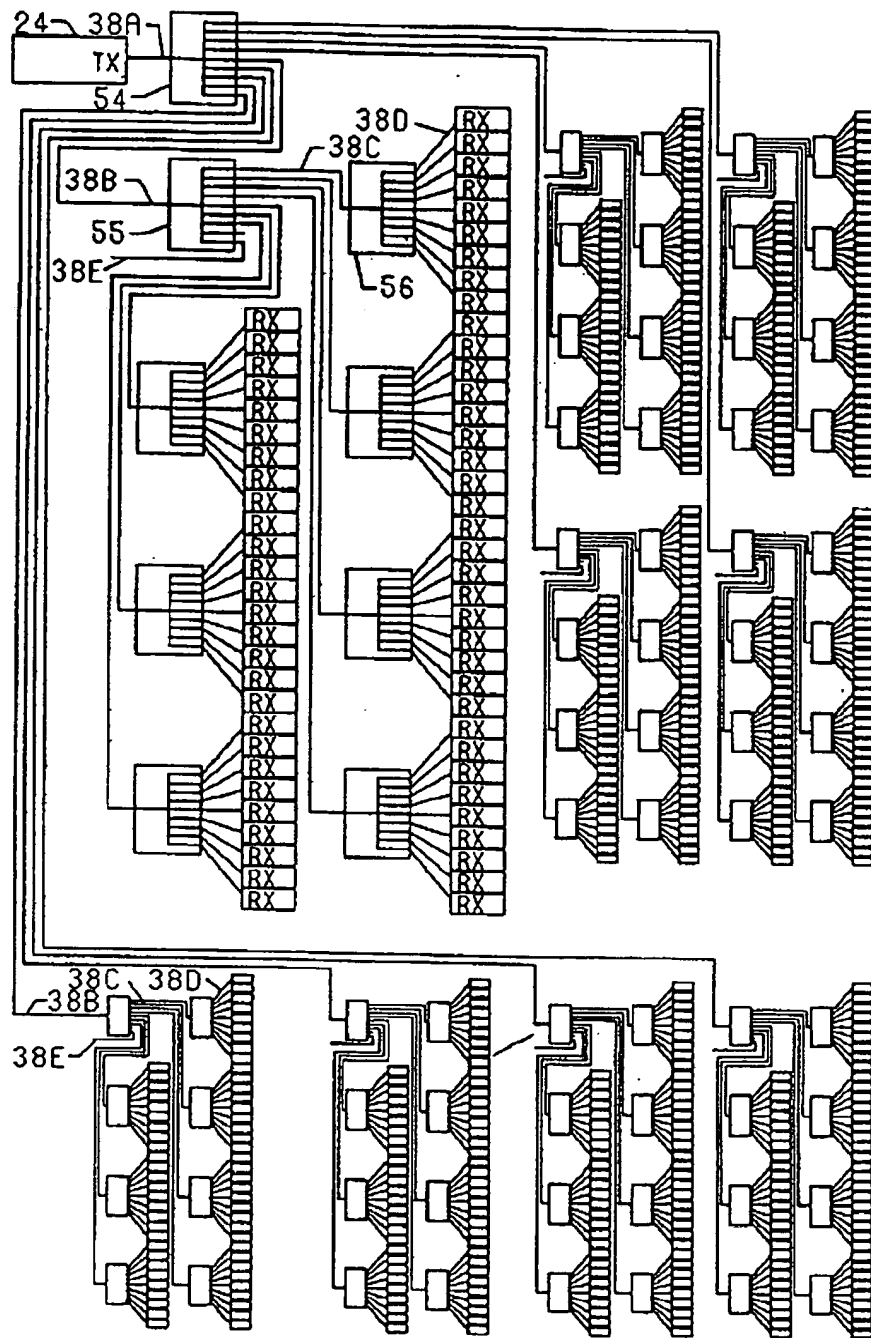


FIG.19

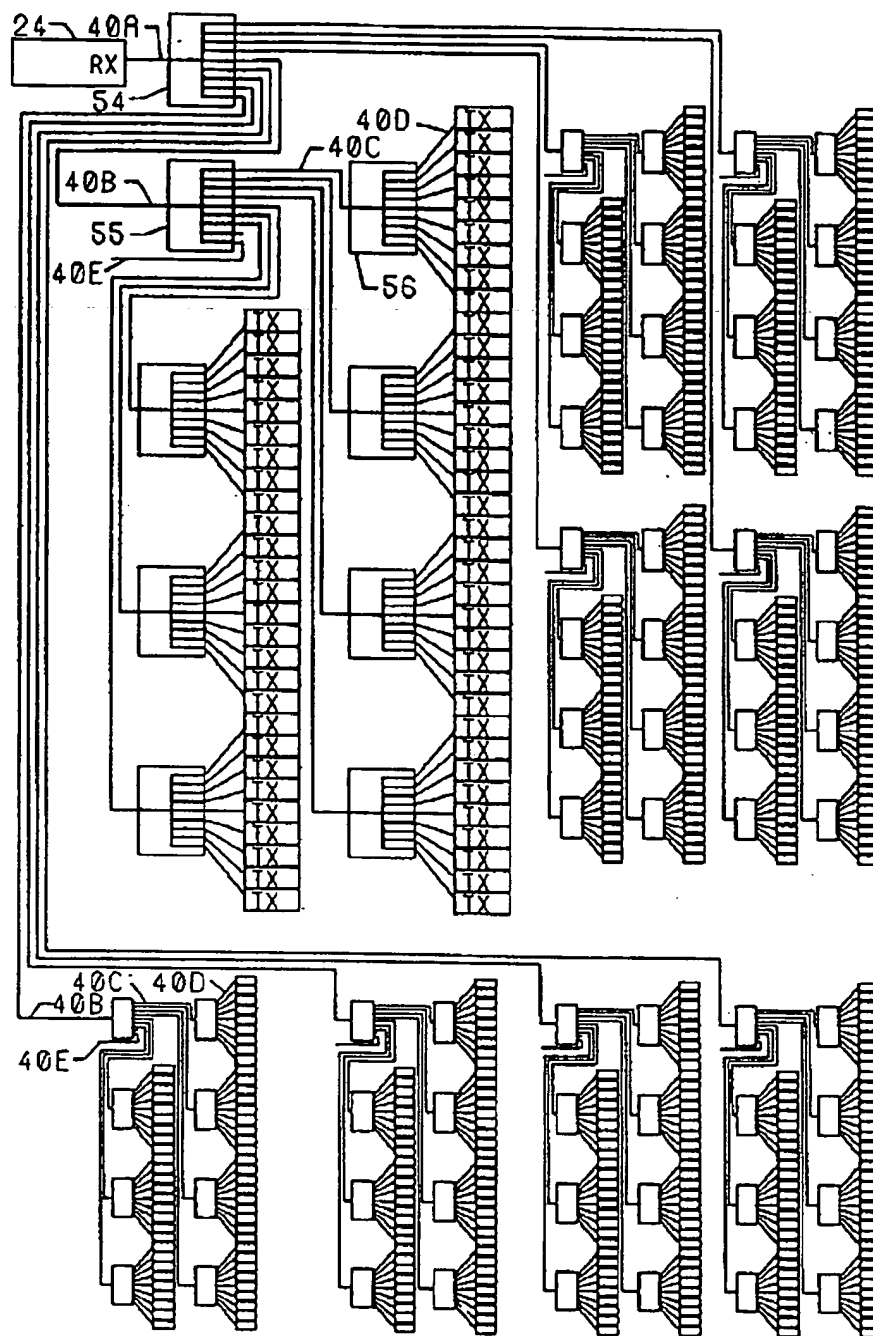


FIG. 20

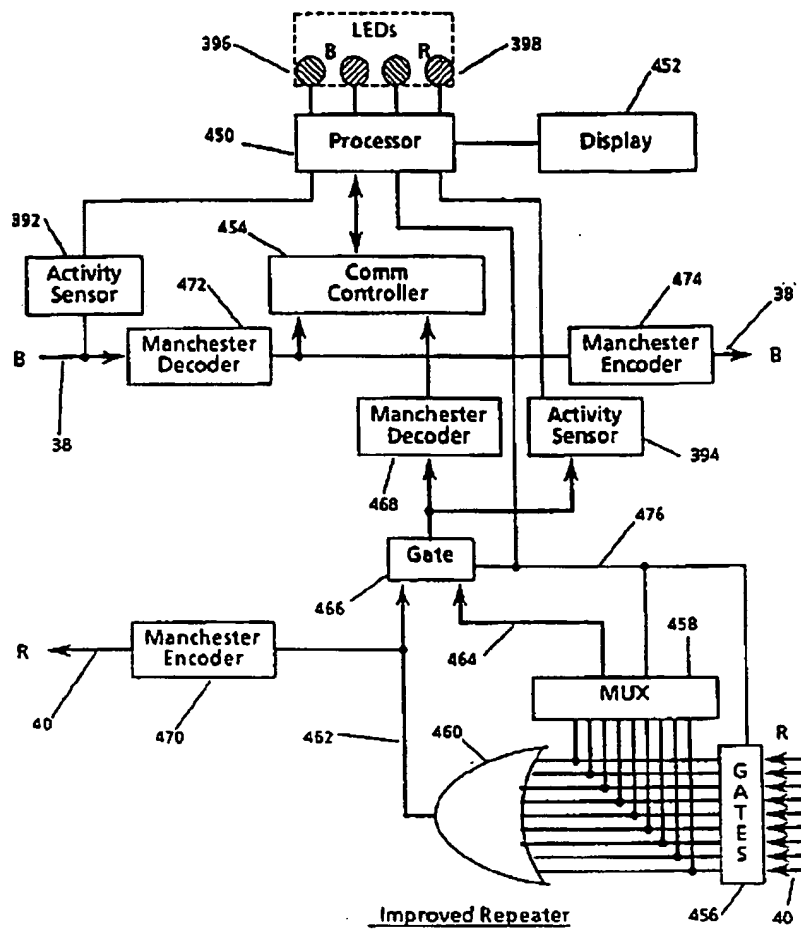


FIG.22

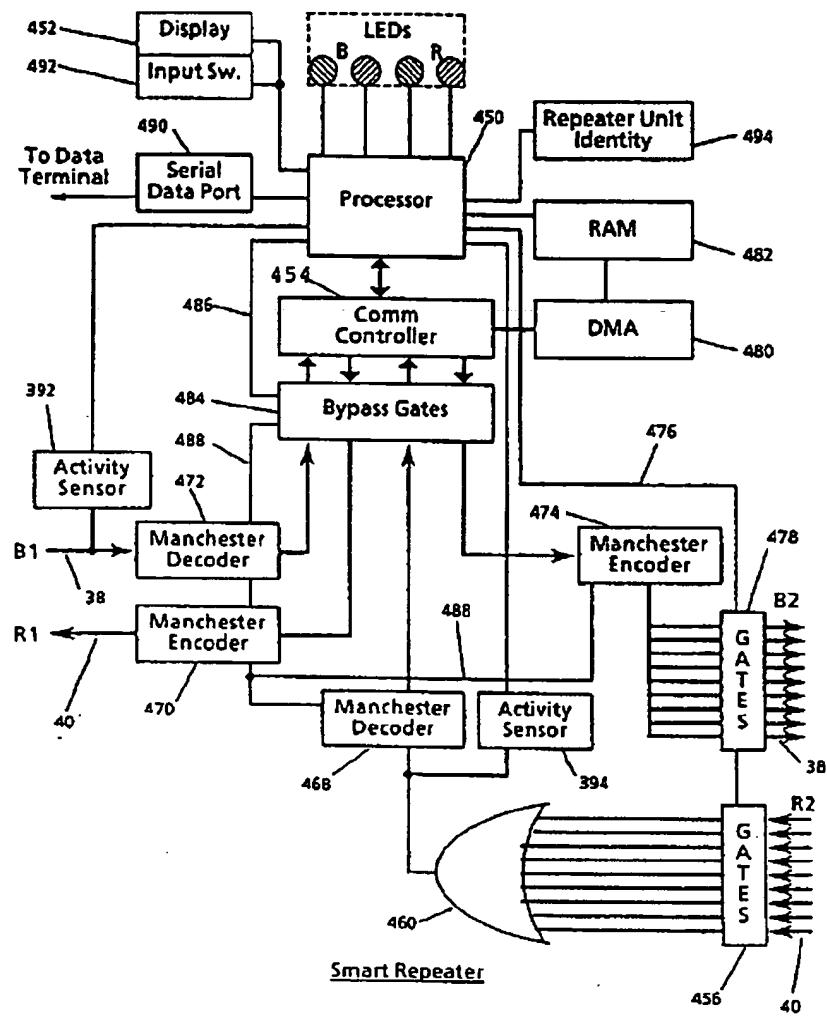


FIG.23

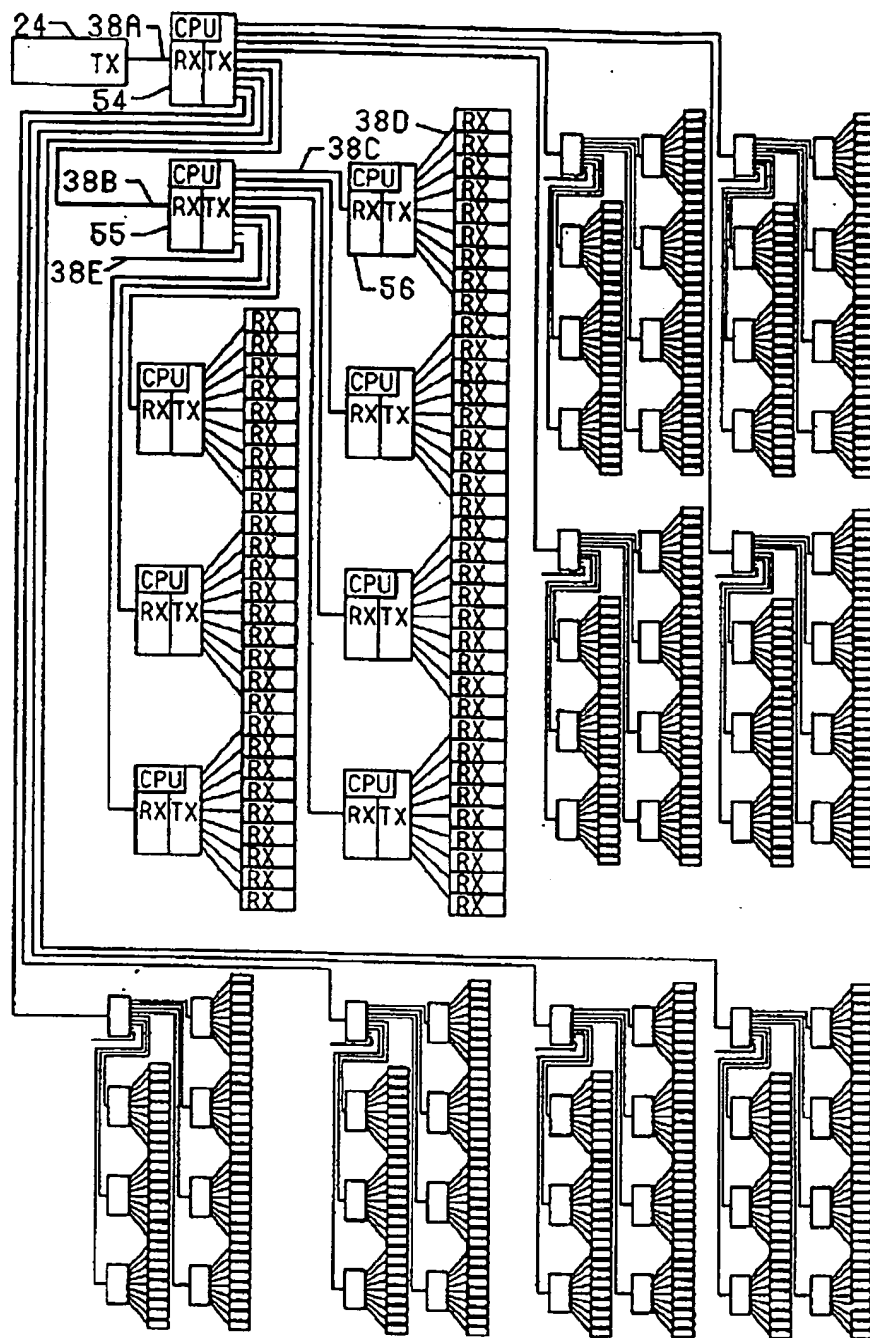


FIG.24

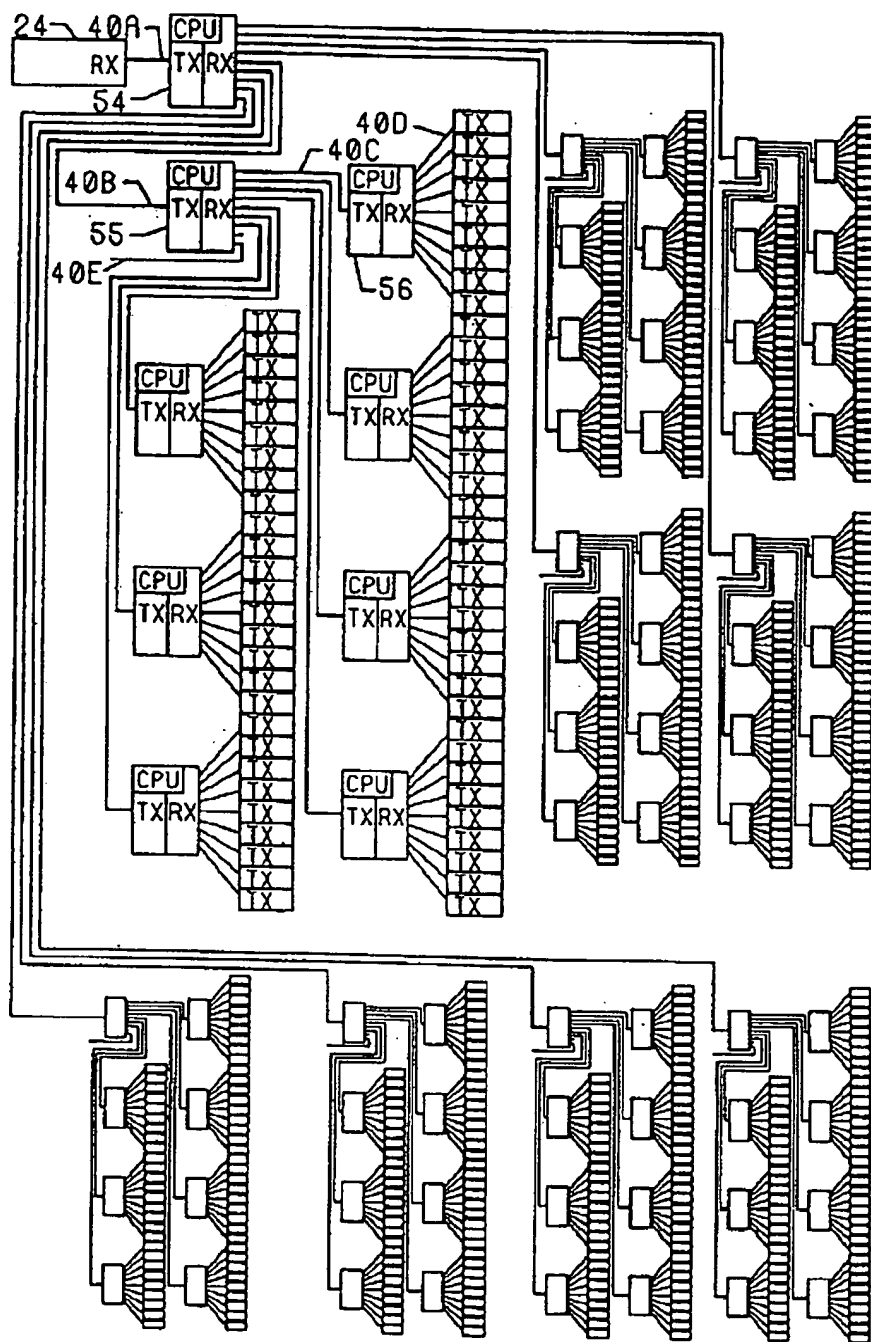


FIG. 25

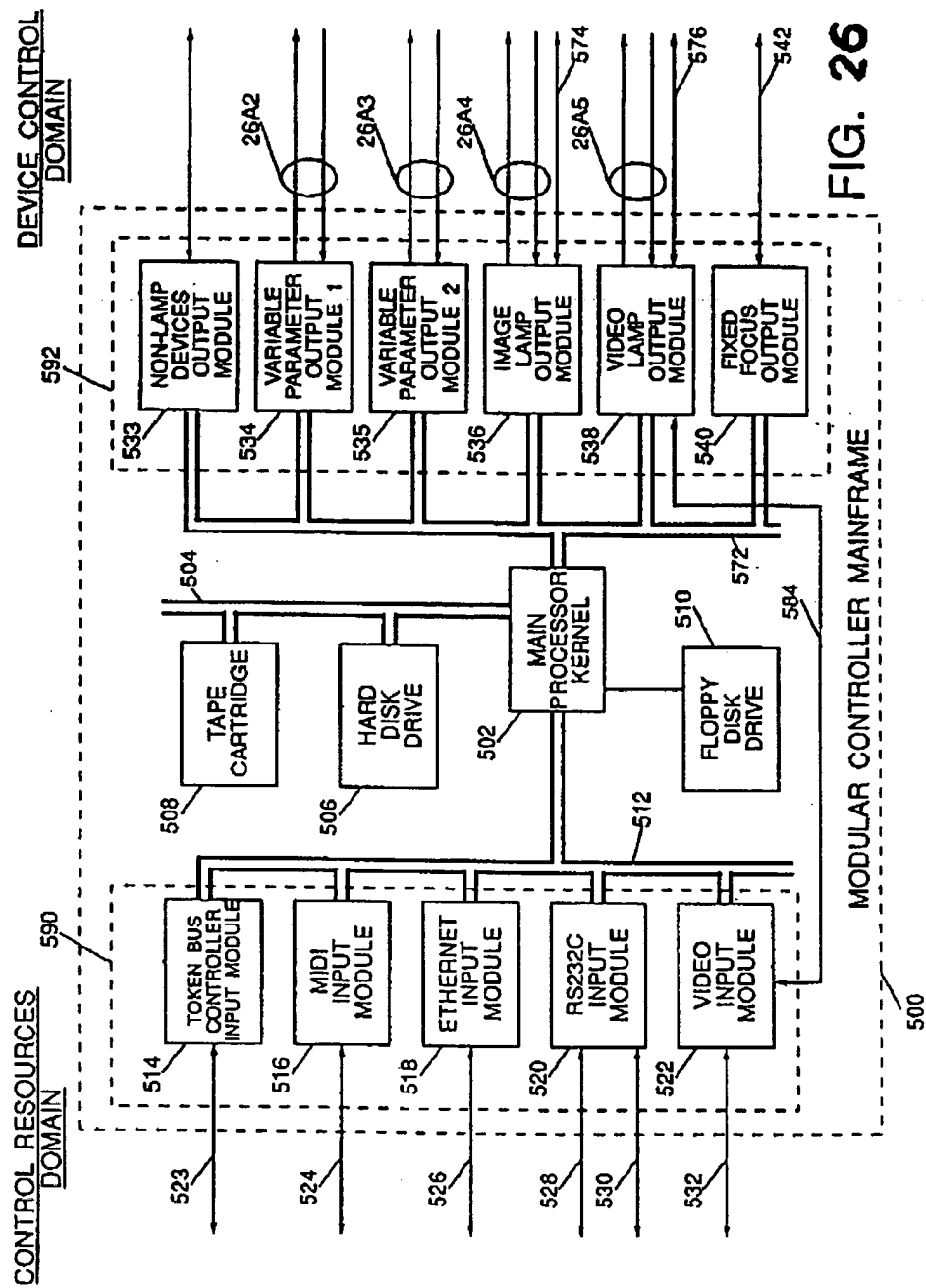


FIG. 26

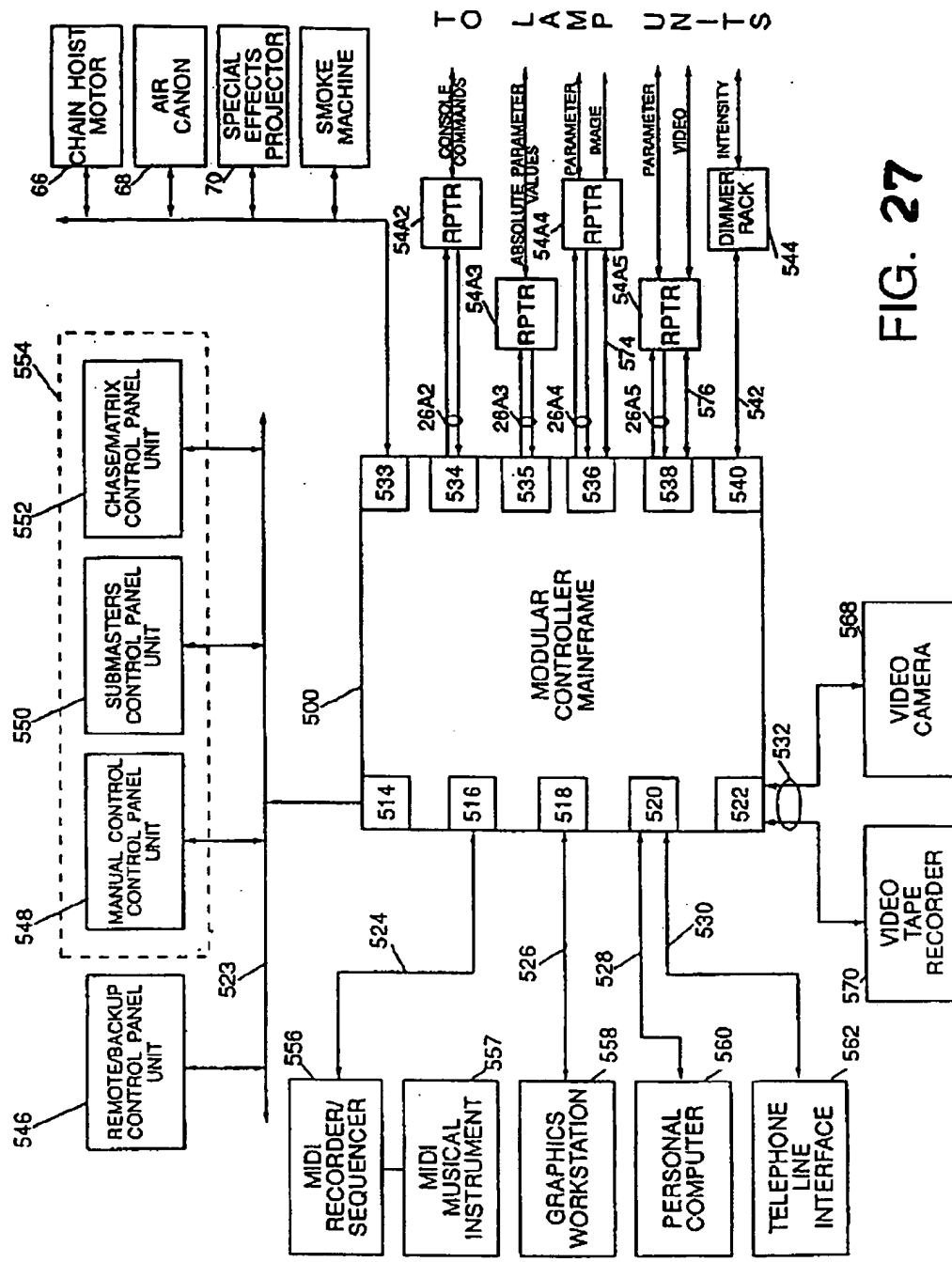


FIG. 27

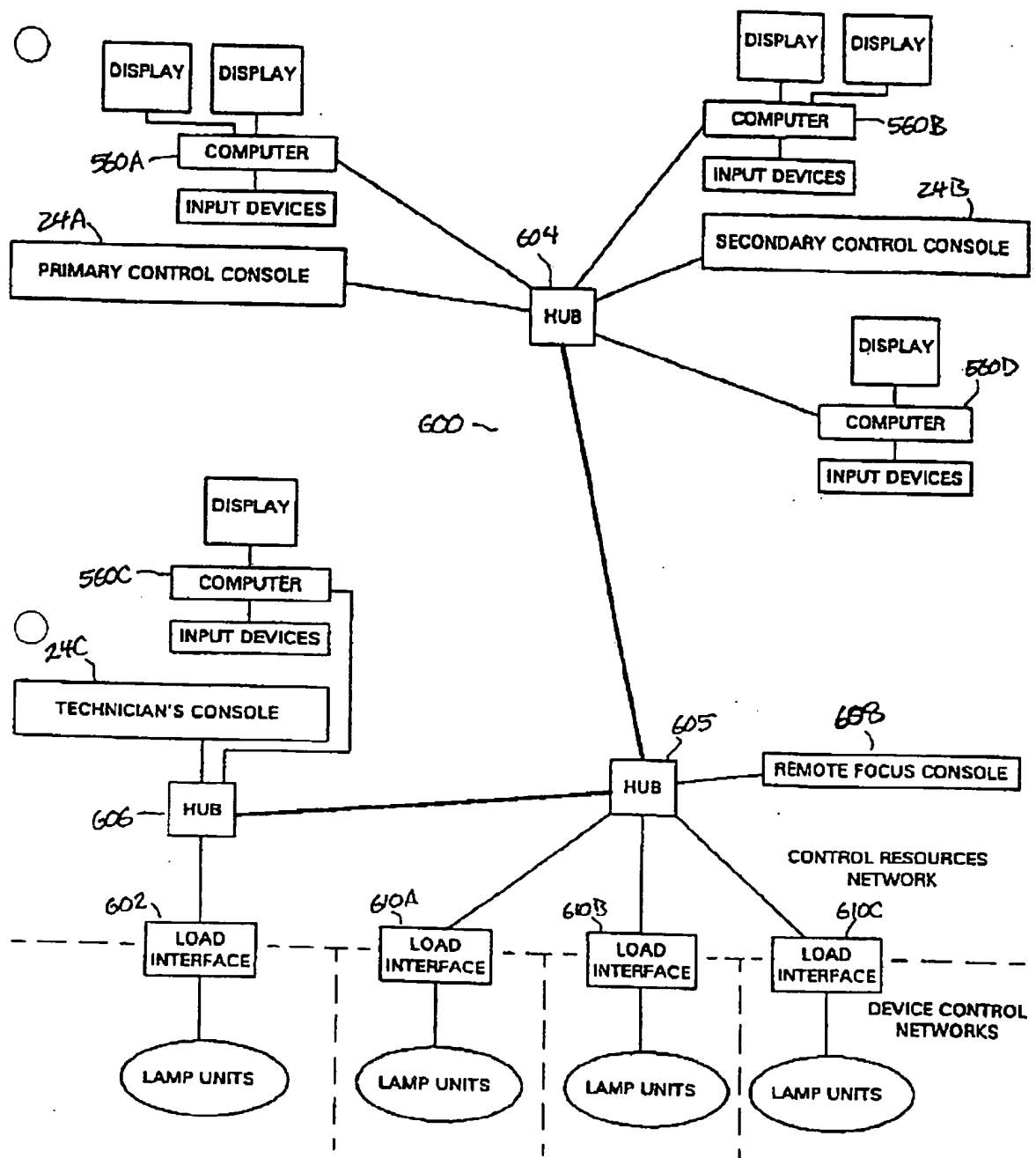


FIG.28

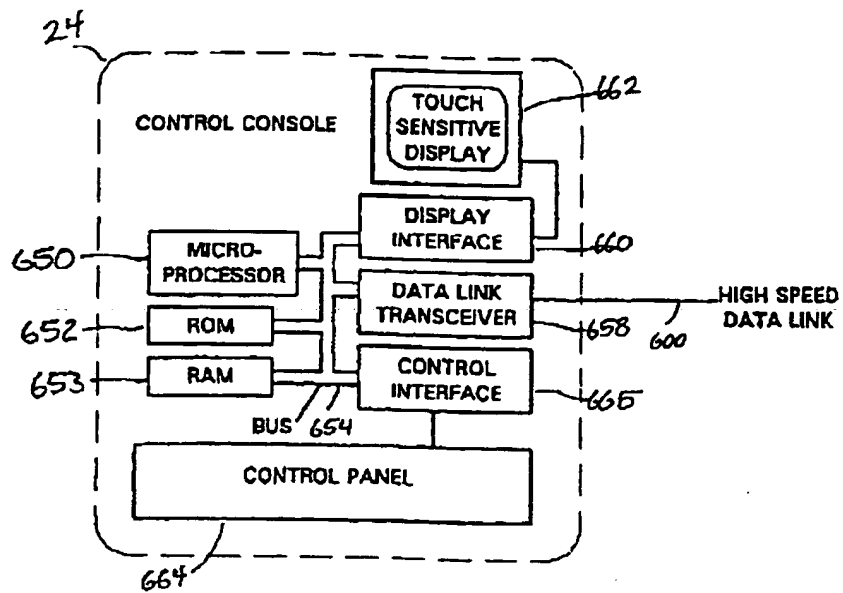


FIG. 29

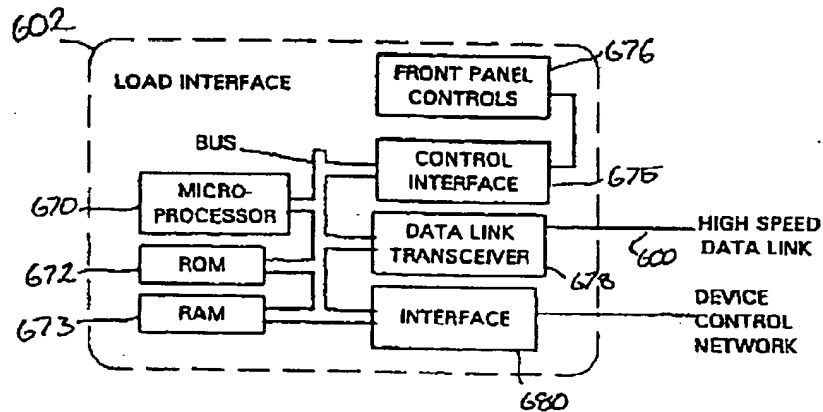


FIG. 31

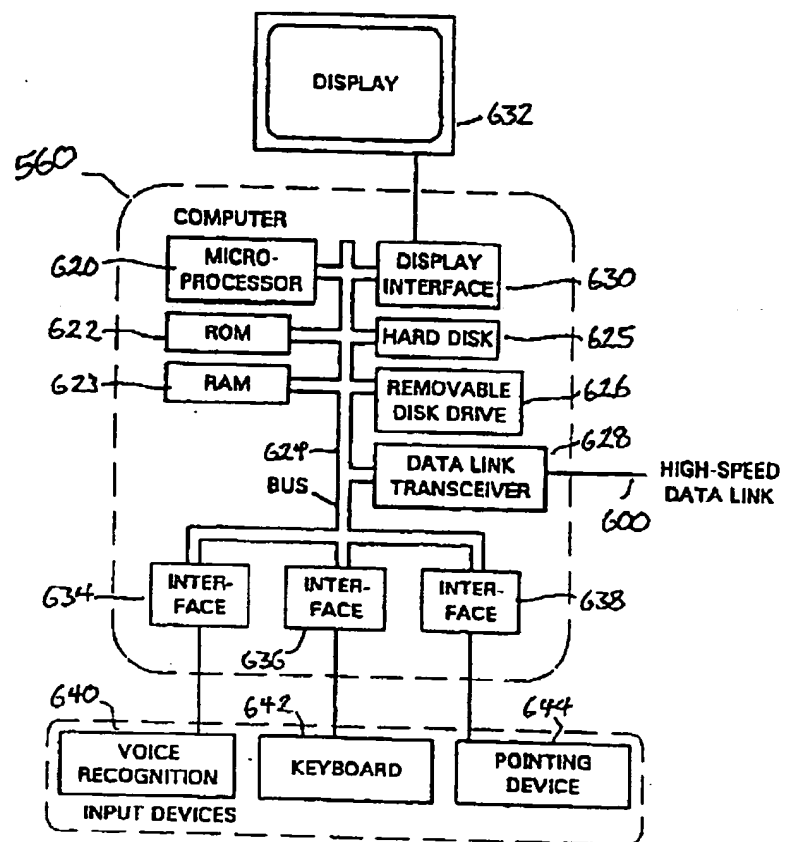


FIG.30

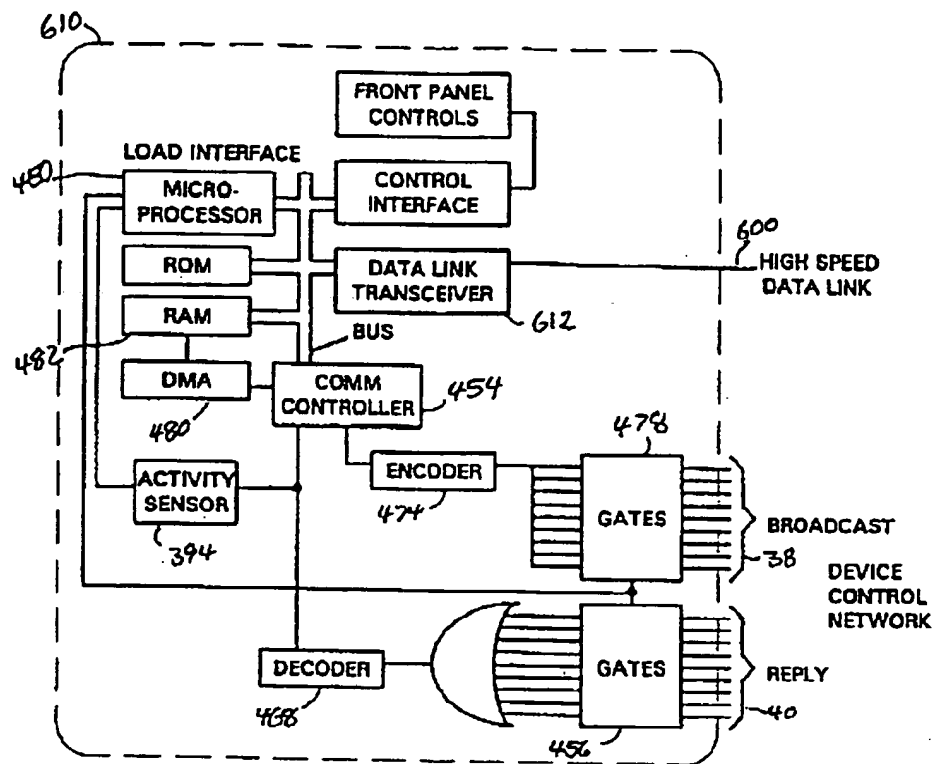


FIG.32

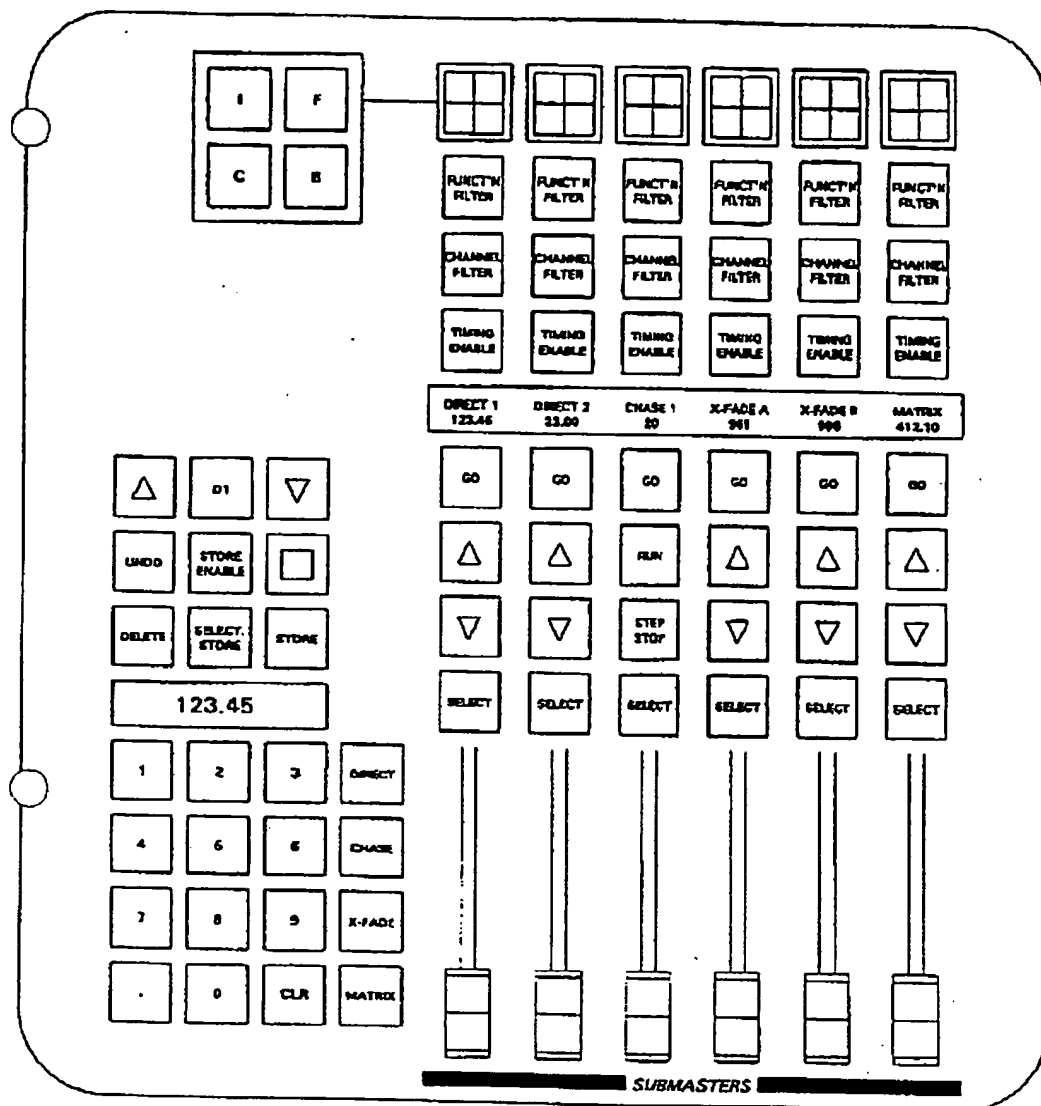


FIG. 33

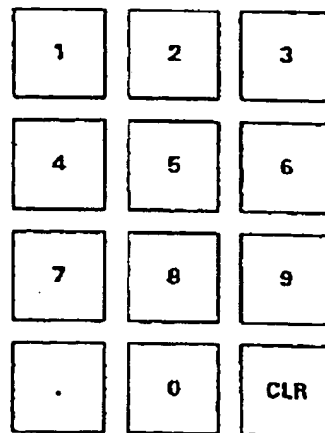
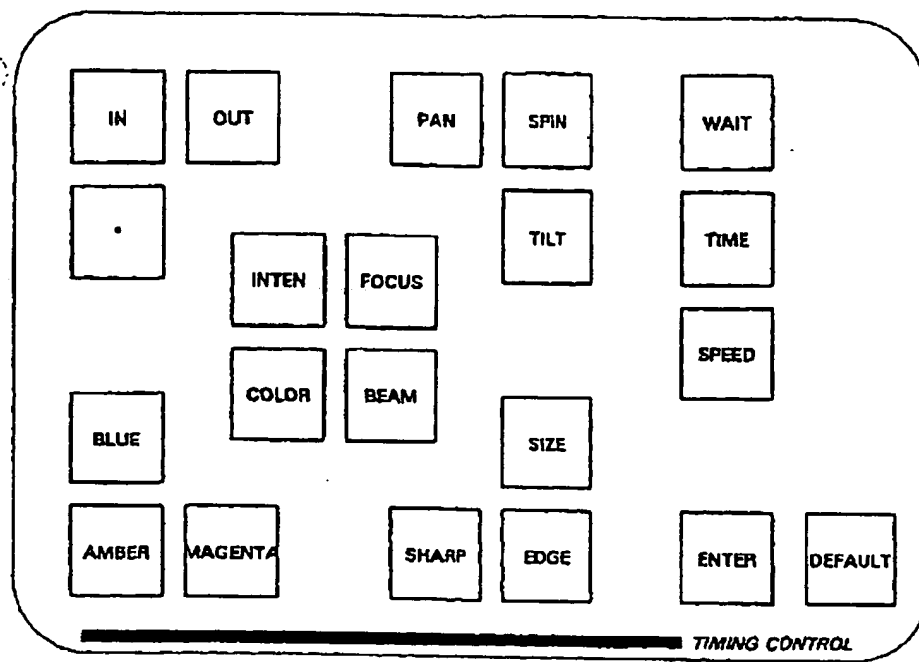


FIG.34

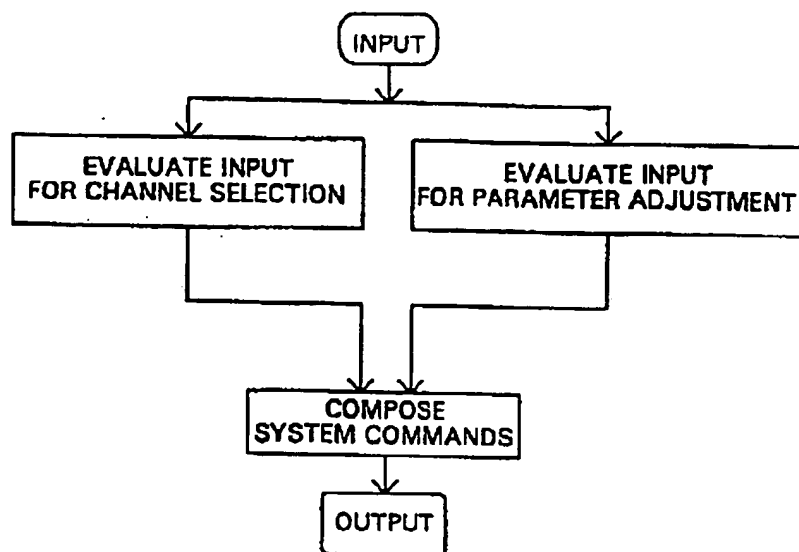


FIG.35

1. Abstract

[Problems]

A distributed control system for a lighting system, including: one or more control devices for entering parameter-controlling inputs according to a specified format, the parameter-controlling inputs directing the operation of the lighting system, the control devices including a data processor coupled to the parameter-controlling inputs and a memory coupled to the processor; one or more computing devices for storing, editing, and displaying data related to the parameter-controlling inputs, the computing devices including at least a data processor, a memory coupled to the processor, and a data display device coupled to the processor; one or more load interface modules each including a data processor for controlling the respective interface module and for monitoring data link signals, each of the load interface modules supporting at least one device-control data link network; a control-resources data link network connecting the control devices, the computing devices, and the load interface modules; and at least one device-control data link network having a common path for connecting the load interface module to a plurality of multiple-parameter lamp units having a plurality of adjustable parameters relating to beam characteristics and a driver for controlling a plurality of the parameters in response to the parameter-controlling inputs.

2. Representative Drawing

F i g . 2

|

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.